Total Solution for Industrial Automation

CIMON PLC – GUIA RAPIDA DE INICIO

Cimon PLC
Guía rápida de inicio

Version: 1.0 Abril 2012
Copyright © Alk357

KDT System
https://www.cimon.co.kr/eng/

KDT SYSTEMS 주)케이디티 시스템즈 KDT SYSTEMS CO.,LTD.

www.kdtsys.com

Tabla de contenido

INDICE 2

1. SOFTWARE DE PROGRAMACION DEL PLC CIMON 4
1.1 INSTALACION DEL SOFTWARE CICON 5
1.1.1 Pasos de instalación del software 5
1.1 ENTORNO DE PROGRAMACION DEL SOFTWARE CICON 6
1.1.1 Barra de menú principal y herramientas 6
1.2.2 Árbol de proyecto 6
1.2.3 Área de edición de Programa 7
1.2.4 Administrador de variables 8
1.2.5 Ventana de mensajes 8
2. CREACION DE UN PROYECTO 9
3. CREACION DE UN PROGRAMA BASICO 11
3.1 Creación de un programa en lenguaje Ladder. 11
3.2 Inserción de contactos de entrada y salida sobre programa en Ladder 12
4. COMPILACIÓN Y DESCARGA DE PROGRAMA HACIA PLC 22
4.1 Compilación de programa 22
4.2 Descarga de programa hacia PLC 23
4.2.1 Procedimiento para descargar programa del PLC vía el puerto USB) 23
4.2.1 Conexión del PLC vía el puerto Serial 26
4.2.1 Conexión del PLC vía el puerto Ethernet 26
5 COMUNICACIÓN EN LÍNEA CON EL PLC 27
5.1 Conexión en línea en modo monitor 27
5.2 estado de operación de PLC 28
6. TIPOS DE VARIABLE Y MEMORIA INTERNA 29
6.1 Lista de variables de memoria 29
6.2 Nomenclatura de variables en programa 30
7. MANEJO DE INSTRUCCIONES BASICAS LADDER 32
7.1 Instrucciones tipo flanco de bit 32
7.2 Instrucciones tipo Bit Set / Reset 32
7.3 Instrucciones de movimiento y copia de datos 33
7.4 Instrucciones aritméticas 34
7.4.1 Instrucciones de suma y resta 34
7.4.2 Instrucciones de multiplicación y división 34
7.5 Instrucciones de Comparación 36
7.6 Instrucciones de Temporizadores y Contadores 37
7.6.1 Temporizadores 37
7.6.2 Contadores 39
7.7 Instrucciones de operaciones lógicas 40
7.8 Llamadas de subrutina 41
7.6.1 instrucción ECALL 42
7.6.2 Creación de programa subrutina 43
7.9 Instrucciones de salida con pulso 44

8. MANEJO DE MODULOS DE EXPANSION 45
8.1 Lectura de módulos Digitales de expansión. 46
8.2 Lectura de módulos análogos. 49
8.2.1 Instrucción de lectura de datos de Módulos de expansión. 49
8.2.2 Instrucción de escritura de datos sobre Módulos de expansión. 50
8.2.3 Lectura de entrada análoga sobre modulo. 51
8.2.4 Escritura sobre salida análoga de modulo. 53
9. COMUNICACIÓN CON DISPOSITIVOS EXTERNOS 54
9.1 Comunicación con el HMI Cimon. 54
9.2 Comunicación utilizando el protocolo MODBUS 55
9.2.1 Comunicación vía MODBUS RTU modo Maestro. 57
9.2.2 Comunicación vía MODBUS TCP modo Maestro. 60
9.3 Creación de un protocolo personalizado. 63
9.3.1 Ventana de Edición y creación de tramas 65
9.3.2 Creación de tramas de Transmisión/recepción de datos 66
10. FUNCIONES ESPECIALES INCORPORADAS 71
10.1 BLOQUE CONTROL PID CON EL PLC CIMON. 71
10.2 Parámetros Básicos dentro del control PID. 72
10.3 Formula de salida para el control PID en función de sus parámetros 74
10.4 Rango de valores máximos para parámetros PID 75
10.5 Distribución de registros de memoria en un Bloque de control PID 75
10.5.1 PIDINIT starting area (PIDINIT Start) 76
10.5.2 PIDCAL starting area(PIDCAL Start) 77
10.6 Graficas de tendencias con control PID 77
11. CONEXIONES ELECTRICAS EN LOS PLCs 78
11.1 Descripcion de puertos fisicos de comunicación. 79
11.1.1 Conexión del puerto serial: 79
11.1.2 Conexiones sobre el puerto Ethernet: 81
11.1.3 Conexiones sobre el puerto USB 82
11.2 Conexión de las entradas y salidas Digitales 82
11.3 Conexión de las entradas y salidas Análogas 86
11.4 Especificaciones tecnicas 88

1. SOFTWARE DE PROGRAMACION DEL PLC CIMON

El software de programación de las de los PLC Cimon, es el CICON, En este entorno de Programación se pueden programar la series de PLC CIMON: CP, XP, PLCs, BP.

A través de este software es posible la programación, la depuración, la simulación y las funciones de diagnóstico del sistema sobre todas las aplicaciones requeridas y programadas en los PLC.

La conexión en línea con las series de PLC Cimon, puede realizarse de diversas maneras utilizando los puertos de comunicación de la computadora tales como, Serial, Ethernet, RS232, sin necesidad de algún tipo de cable especial de fabricación en pedido.

Este software es gratuito y de libre distribución, puede ser descargado desde la página web de KDT systems (Customer Support), o pedido a solicitud gratuita a su distribuidor Local.

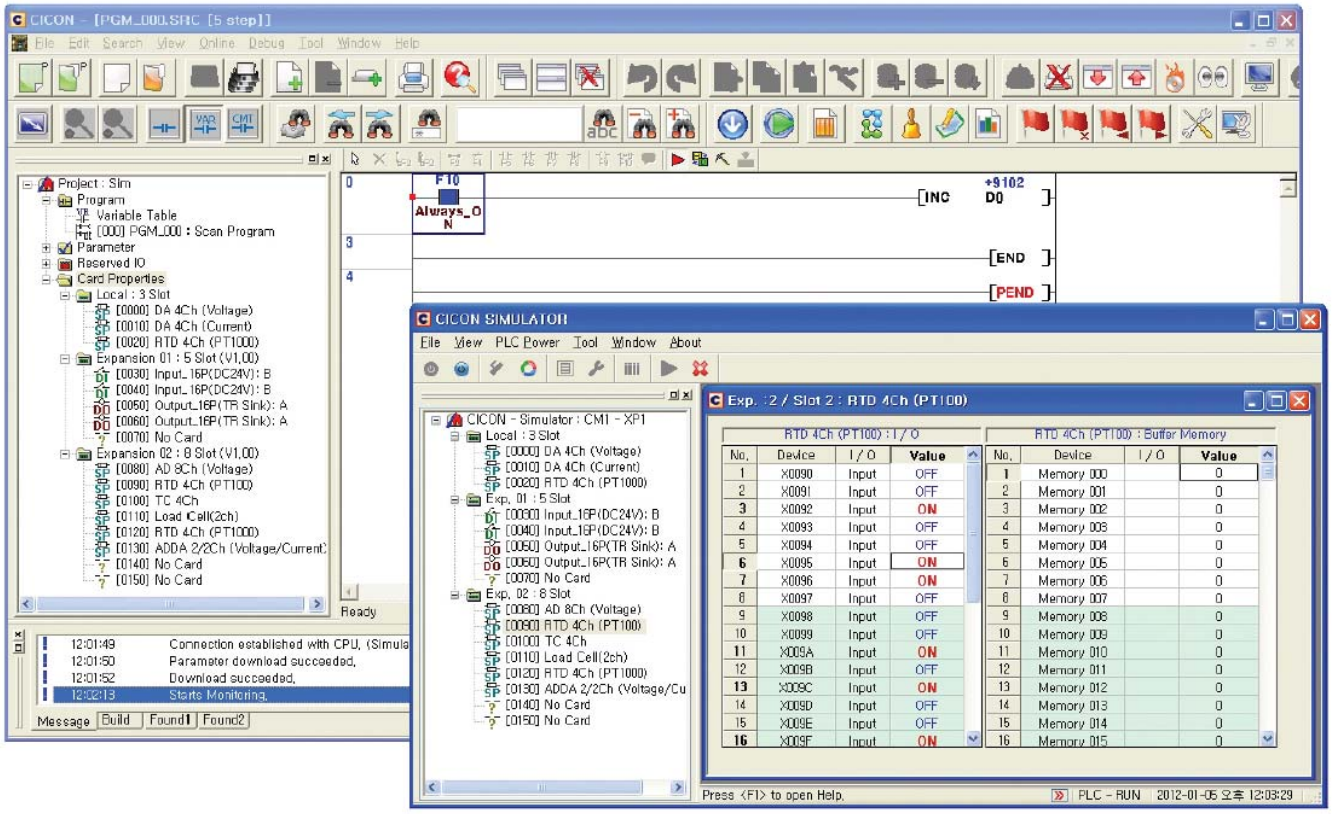


Fig1: Software CICON

1.1 INSTALACION DEL SOFTWARE CICON

Dentro de los requisitos recomendados para la computadora donde se instalara el software CICON, son los siguientes:

- Computadora con CPU Pentium 133Mhz o Superior, 64MB de RAM o más incluyendo la expansión de memoria.
- 1 Puerto serial RS232 es requerido cuando se requiera conectar a un PLC mediante su puerto o módulo de comunicación RS232C/422/485.
- 1 Puerto Ethernet es requerido para conectarse vía el modulo Ethernet del PLC.
- 1 Puerto USB es requerido para comunicarse directamente con los PLC de la serie PLCs.
- Disco Duro con al menos 13MB libre de espacio es requerido para la instalación de los Componentes del software CICON, para que el software funcione apropiadamente.
- Compatible con Windows 98SE/2000/XP/7

1.1.1 Pasos de instalación del software

La instalación del software CICON es simple y no requiere conocimientos avanzados de computación, la licencia de uso es libre, a continuación se muestran la guía de pasos realizados para este procedimiento.

- Doble clic en el instalador del programa
- Luego aparecerá una ventana para la selección del idioma usado en los mensajes mostrados en el proceso de instalación.
- Clic en OK, para luego seleccionar la carpeta de instalación del software, se recomienda utilizar la dirección por defecto, clic en el botón “Next” (siguiente).
- Si desea personalizar la dirección de carpeta destino del software CICON, clic en el botón [Browser] para luego seleccionar una nueva carpeta de instalación del programa.
- Luego de continuar el proceso, el procedimiento de instalación es automático.

1.2 ENTORNO DE PROGRAMACION DEL SOFTWARE CICON

El entorno de programación está distribuido principalmente en 5 áreas o bloques:

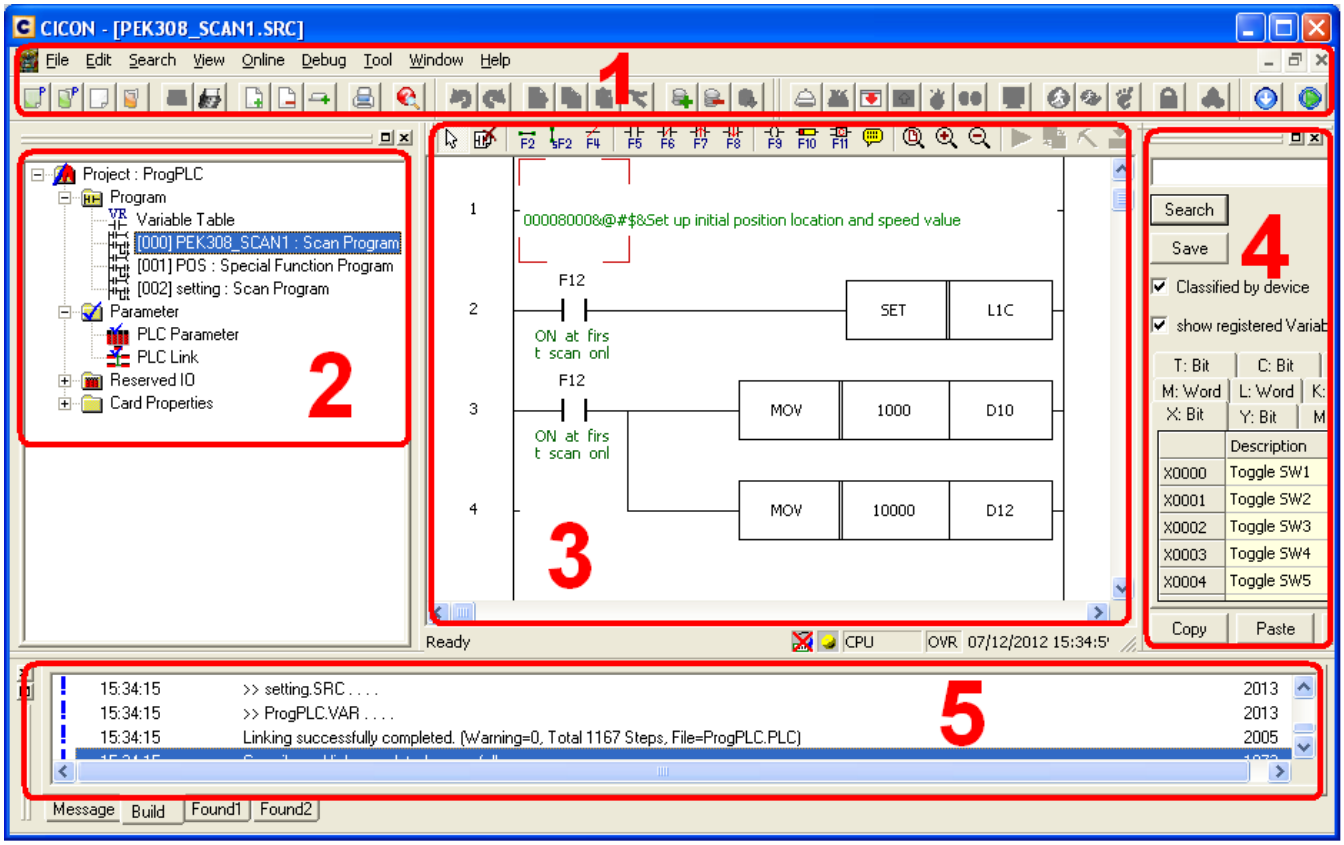


Fig2: Entorno de programación del software CICON.

- Barra de menú principal y herramientas (1)
- Árbol de proyecto (2)
- Área de edición de Programa (3)
- Administrador de variables (4)
- Ventana de mensajes (5)

1.2.1 Barra de menú principal y herramientas

Mediante este bloque podemos acceder a todas las funciones y herramientas implementadas mediante el software CICON. Funciones como creación de proyecto, edición de programas, Conexión en línea con el PLC, Compilación y descarga de programa, etc.

1.2.2 Árbol de proyecto

En este bloque se podrán visualizar la lista de programas, parámetros, y ajustes relacionados al proyecto actual a programar sobre el PLC. En este bloque también se podrá observar la lista de módulos interconectados sobre el CPU principal (PLC).

1.2.3 Área de edición de programa

Sobre esta área se podrá visualizar el código de programado para las funciones y tareas que ejecutara el PLC.

Dentro del Software CICON, se puede utilizar dos tipos de lenguaje de programación: Ladder y Lista de instrucciones.

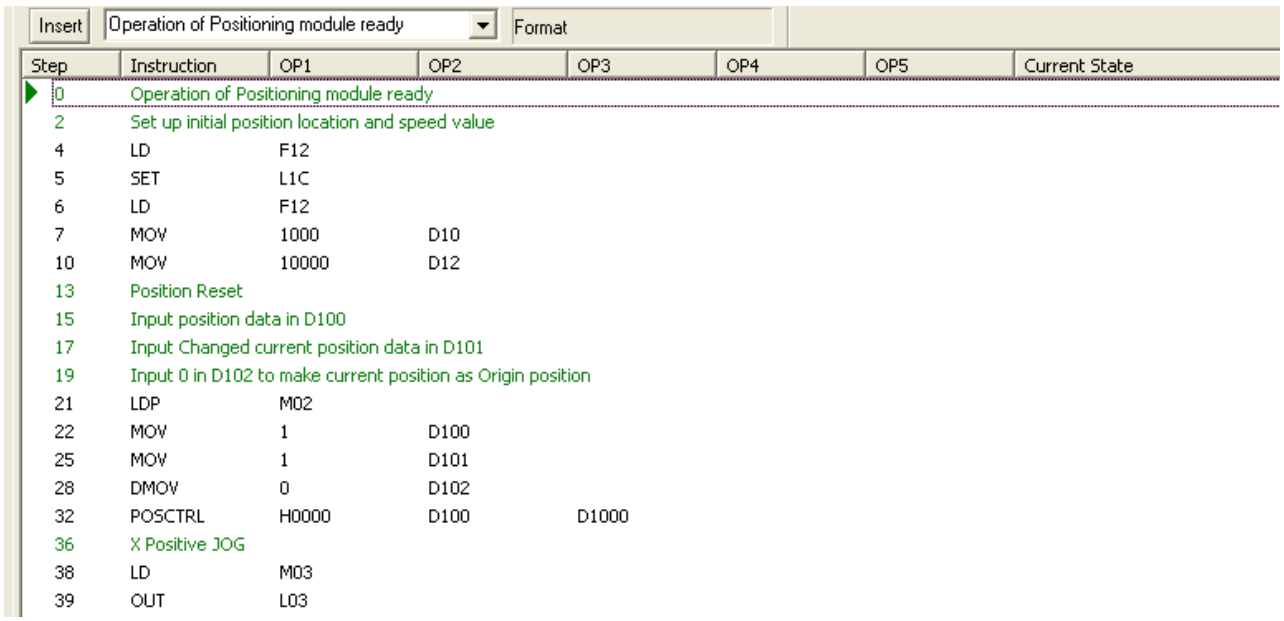

Insert
Operation of Positioning module ready
Format
Step Instruction OP1 OP2 OP3 OP4 OP5 Current State
0 Operation of Positioning module ready
2 Set up initial position location and speed value
4 LD F12
5 SET L1C
6 LD F12
7 MOV 1000 D10
10 MOV 10000 D12
13 Position Reset
15 Input position data in D100
17 Input Changed current position data in D101
19 Input 0 in D102 to make current position as Origin position
21 LDP M02
22 MOV 1 D100
25 MOV 1 D101
28 DMOV 0 D102
32 POSCTRL H0000 D100 D1000
36 X Positive JOG
38 LD M03
39 OUT L03

Fig3: ventana de programación en modo de lista de instrucciones

Existen 2 modos de visualización de la programación en lenguaje Ladder, Nombrados como OLD Style (lado izquierdo de figura 4) y Default Style(lado derecho de figura 4).

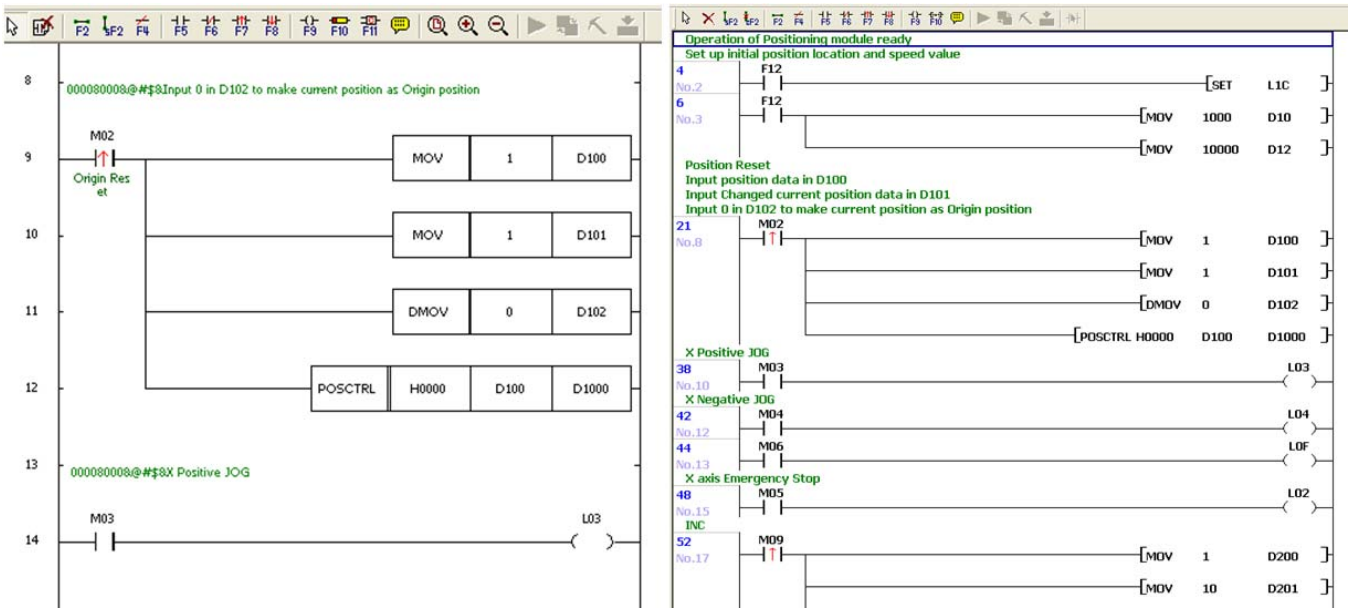


Fig4: modos de visualización del lenguaje Ladder en el software CICON.

- Para alternar la visualización entre modo lista de instrucciones y Ladder, dirigirse a la barra de menú principal, *Tool / IL-LD Conversion*

- Para cambiar el modo de visualización del editor Ladder para el modo "OLD style", dirigirse a la barra de menú principal, *Tool / CICON Options…* Luego aparecerá una ventana (ver figura 5) , Clic sobre la pestaña *"LD Editor"*, y activar la casilla *"Use The old style LD editor"*

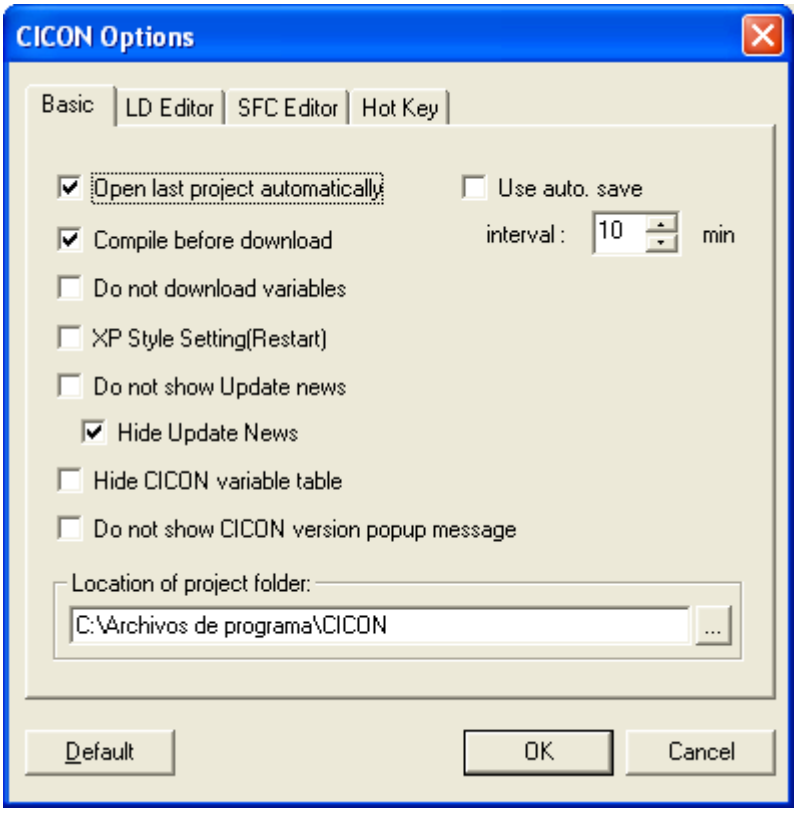


Fig5: ventana de opciones de CICON

1.2.4 Administrador de variables

Sobre este bloque se podrá visualizar todas las variables declaradas sobre el proyecto actual, además de clasificar y separar las variables según el tipo de formato, también nos permite la opción exportar dicho orden de variables en un archivo CSV, visualizado en Excel.

1.2.5 Ventana de mensajes

En esta ventana se podrá Visualizar y registrar automáticamente todos los eventos generados a través del software CICON, en la administración y edición del proyecto.

Esta ventana es muy útil para verificar el estado de las funciones y tareas ejecutadas en el entorno del software CICON.

2. CREACION DE UN PROYECTO

A continuación se listara los pasos necesarios para la creación de un proyecto básico dentro del software CICON.

Paso1: Ejecutamos la aplicación CICON, por defecto puede ser accedida desde el menu de inicio de Windows, *Todos los programas / Cimon /CICON*

Paso2: Una vez abierto el software CICON, podemos verificar la versión instalada sobre nuestra computadora, para ello Ir al menú principal, *Help / About…*

Fig6: Ventana de version instalada del CICON.

Paso3: Para crear un nuevo proyecto damos clic en "New project…", esto sobre el menu principal, *File / New Project*

Paso4: Una vez seleccionada esta opción aparecerá una nueva ventana, (ver figura 7) en esta nueva ventana se ingresara los siguientes datos principalmente:

- Dirección destino de la carpeta que almacenara los archivos del proyecto creado *"Location of Project Folder"*
- Nombre de proyecto, en *"Project properties / Name"*
- El modelo de CPU a utilizar, *" Projects Properties / CPU Type".* las series disponibles para la programación en CICON, se detalla en menú de ayuda Cicon. Para el ejemplo mostrado en la figura 7, el PLC elegido es el de la serie PLCs.
- La opción de password para la apertura y modificación del proyecto se puede establecer mediante el botón *"password"*

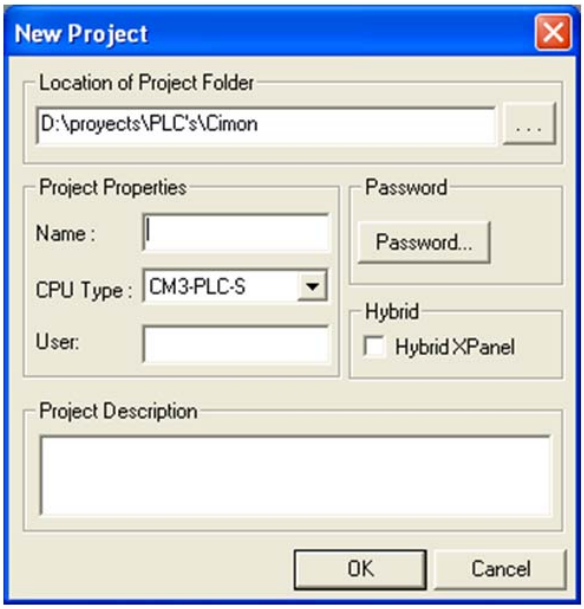


Fig7: Ventana de creacion de nuevo proyecto

- También se puede establecer la opción del nombre de usuario sobre "user" y además alguna nota o descripción referente al proyecto creado "Project Description"

Los parámetros mínimos requeridos para la creación del proyecto son, *"Location of Project Folder", "Project Name" y "CPU Type".*

Una vez introducido los datos requeridos, aparecerá un nuevo proyecto en blanco sobre el entorno de programación del CICON.

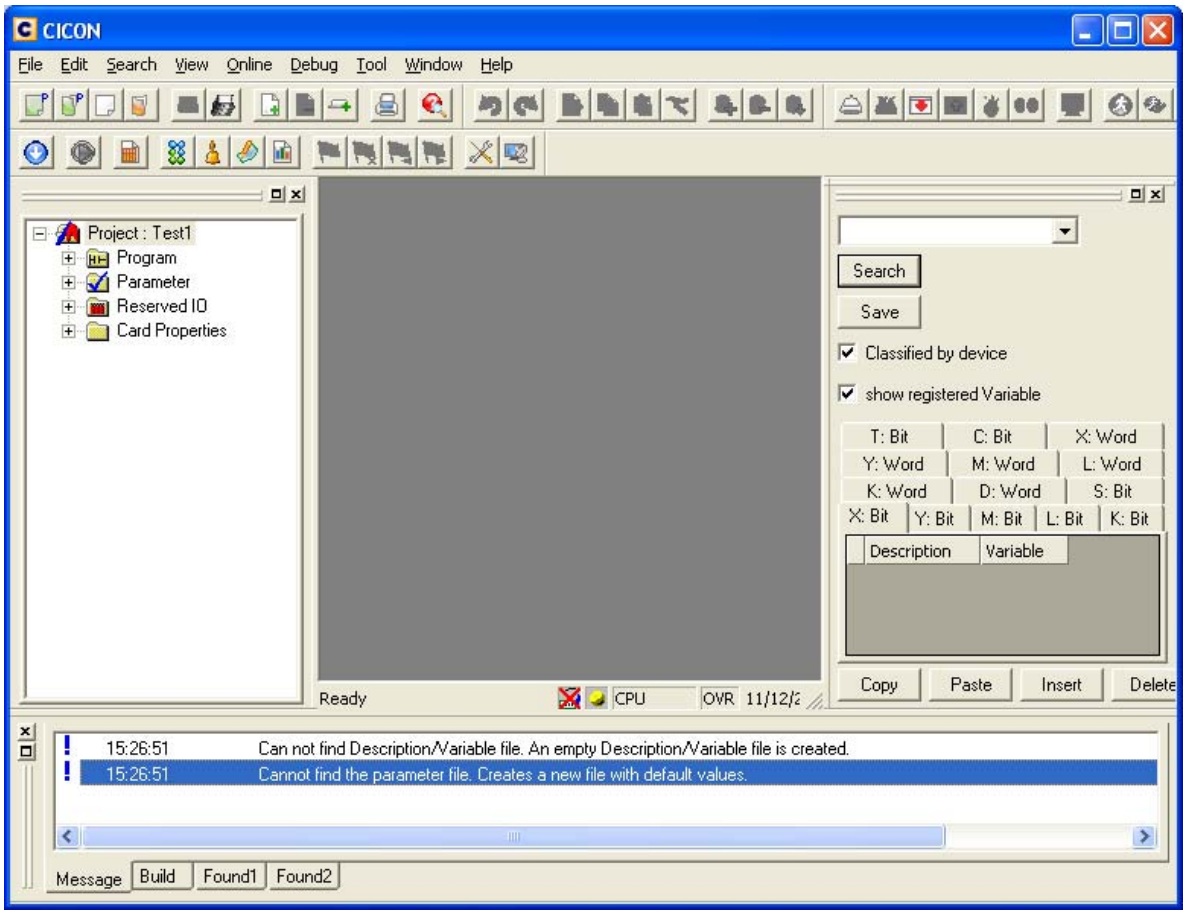

CICON
File Edit Search View Online Debug Tool Window Help
Project : Test1
Program
Parameter
Reserved IO
Card Properties
Search
Save
Classified by device
show registered Variable
T: Bit C: Bit X: Word
Y: Word M: Word L: Word
K: Word D: Word S: Bit
X: Bit Y: Bit M: Bit L: Bit K: Bit
Description Variable
Copy Paste Insert Delete
Ready CPU OVR
15:26:51 Can not find Description/Variable file. An empty Description/Variable file is created.
15:26:51 Cannot find the parameter file. Creates a new file with default values.
Message Build Found1 Found2

Fig8: Nuevo proyecto creado en blanco.

Para verificar que se haya abierto un nuevo proyecto se debe verificar el nombre del proyecto creado o abierto sobre el área del explorador del proyecto.

Sobre la ventana de mensajes se podrá también visualizar el estado de la última acción realizada.

3. CREACION DE UN PROGRAMA BASICO

El Software CICON tiene establecido algunas plantillas de programa que permiten facilitar la programación de una tarea o programa sin necesidad de utilizar un esquema en Ladder.

Sin embargo algunas plantillas de programa también pueden ser convertidas a su equivalente en Ladder.

3.1 Creación de un programa en lenguaje Ladder.

El lenguaje Ladder es el lenguaje de programación Básico dentro de toda las familias y marcas de PLC en el mundo, el editor Ladder del software CICON, presenta las ventajas de su simplicidad y el reducido espacio que pueden ocupar algunas instrucciones muy comunes, facilitando la visualización de varios controles a la vez, y la implementación de una lista larga de Instrucciones.

Paso 1: Antes de crear un programa en Ladder, debemos de seleccionar el tipo de visualización del editor: *"OLD style"* o *"Default Style"*. El tipo de visualización elegido en este ejemplo es el editor "***OLD Style"***.

- Para cambiar el modo de visualización del editor Ladder para el modo "OLD style", dirigirse a la barra de menú principal, *Tool / CICON Options…* Luego aparecerá una ventana (ver figura 5) , Clic sobre la pestaña *"LD Editor"*, y activar la casilla *"Use The old style LD editor"*

Paso 2: Una vez elegido el modo de visualización, dirigirse al menú principal, *File / New Program…* para agregar un nuevo programa al proyecto. Otra forma de agregar un programa al proyecto, es accediendo al explorador del proyecto, dando clic derecho en la carpeta *Program, New Program.*

Paso 3: Al dar clic en *"New Program"*, aparecerá una nueva ventana (ver figura 9) que permitirá seleccionar el tipo de programa a agregar. Para este primer ejemplo seleccionaremos el tipo ***Scan Program***, dentro del Cuadro *Scan Program.*

Adicionalmente también se tiene que agregar el nombre del programa a crear en el cuadro *"Program name"*.

Es posible también agregar una breve descripción en el cuadro de texto *"Remark"*

El Tipo de programa elegido **"Scan Program"** permite crear un programa cíclico que ejecutara constantemente las instrucciones introducidas en el programa creado bajo este formato.

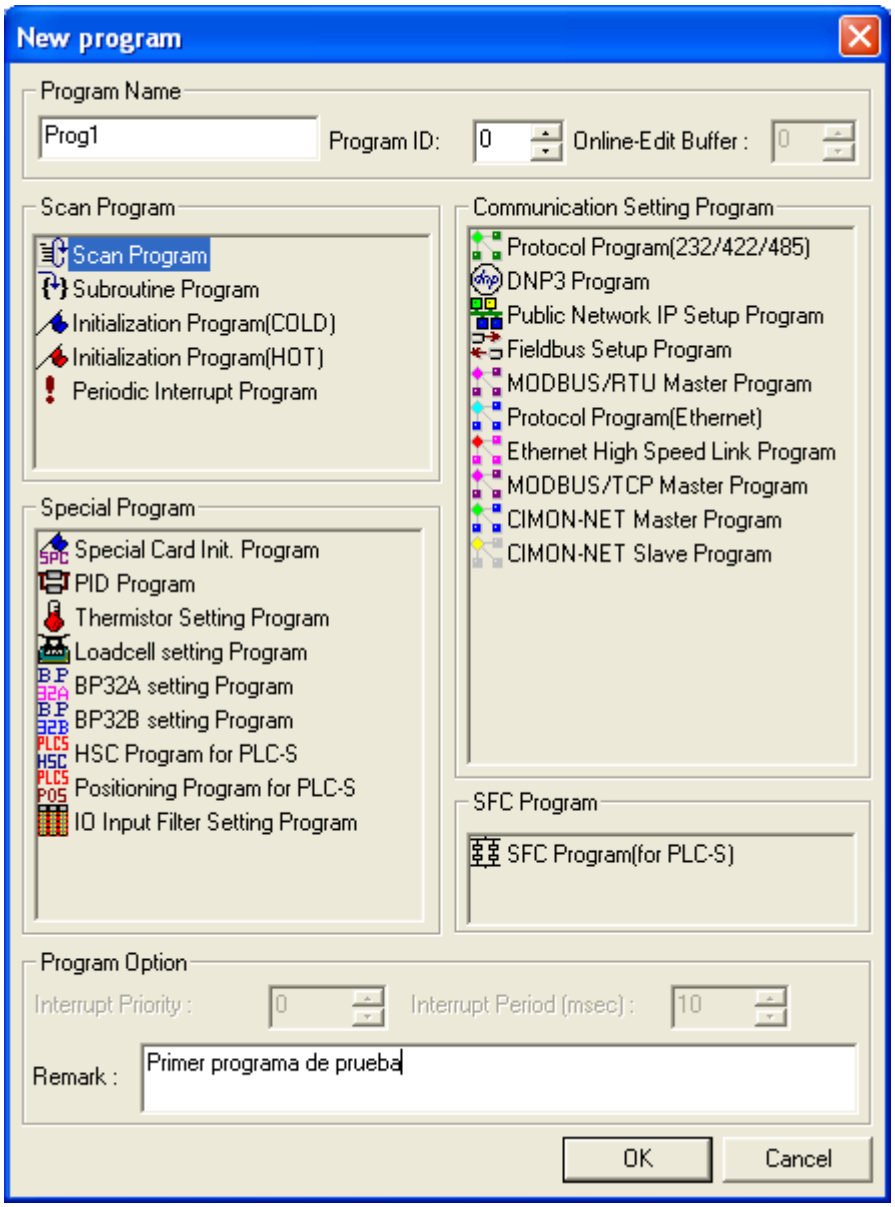


Figura 9: Selección de nuevo programa

Una vez creado el programa este aparecerá sobre el área de edición, además sobre el explorador de proyecto se podrá verificar que un nuevo archivo ha sido agregado.

3.2 Inserción de contactos de entrada y salida sobre programa en Ladder

Para poder ingresar contactos de entrada y de salida utilizamos la barra de herramientas del editor Ladder (ver figura 10), la barra siempre se encuentra en la parte superior del editor Ladder.

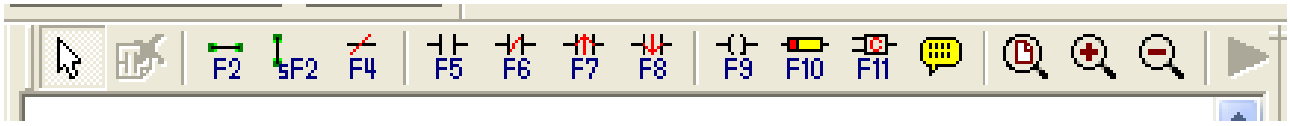


Figura 10: barra de herramientas del editor

A Continuación se realizara un ejemplo de programa en lenguaje ladder, que contendrá la aplicación más básica, encender y apagar contactos de salida.

Paso 1: Seleccionamos el icono marcado en verde de la figura 11, para agregar un contacto de entrada, antes es necesario direccionar el puntero sobre el editor Ladder, donde se ubicara el contacto descrito.

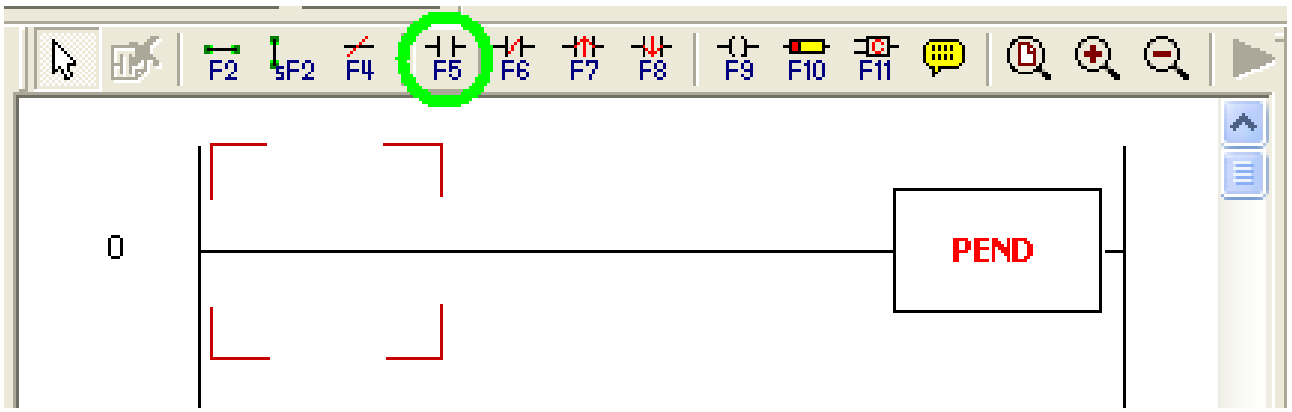


Figura 11: Icono marcado para contactos de entrada.

Paso 2: Al dar clic en el icono marcado de la figura 11 aparecerá una pequeña ventana (figura 12) en el cual escribiremos la dirección de memoria del contacto de entrada. Para nuestro primer ejemplo seleccionaremos “X00” luego dar clic en OK.

Figura12: Ventana de inserción de dirección de contacto de entrada

Paso 3: Luego dirigimos el cursor sobre un área del editor ladder, y damos clic izquierdo ubicando el curso en cuadro rojo en la zona de contactos de salida (ver figura 13).

Paso 4: damos clic izquierdo en un contacto de salida, dando clic izquierdo en el icono sobre el área marcada en verde de la figura 13.

Paso 5: Inmediatamente aparecerá otra pequeña ventana (ver figura 14) donde se introducirá la dirección del contacto de salida a utilizar. Para este ejemplo ingresamos la dirección “Y00” y damos clic en OK.

- Para tener mayor referencia sobre las direcciones de contactos de entrada y salida a utilizar, dirigirse al tema 6, Tipos de variable y memoria interna.

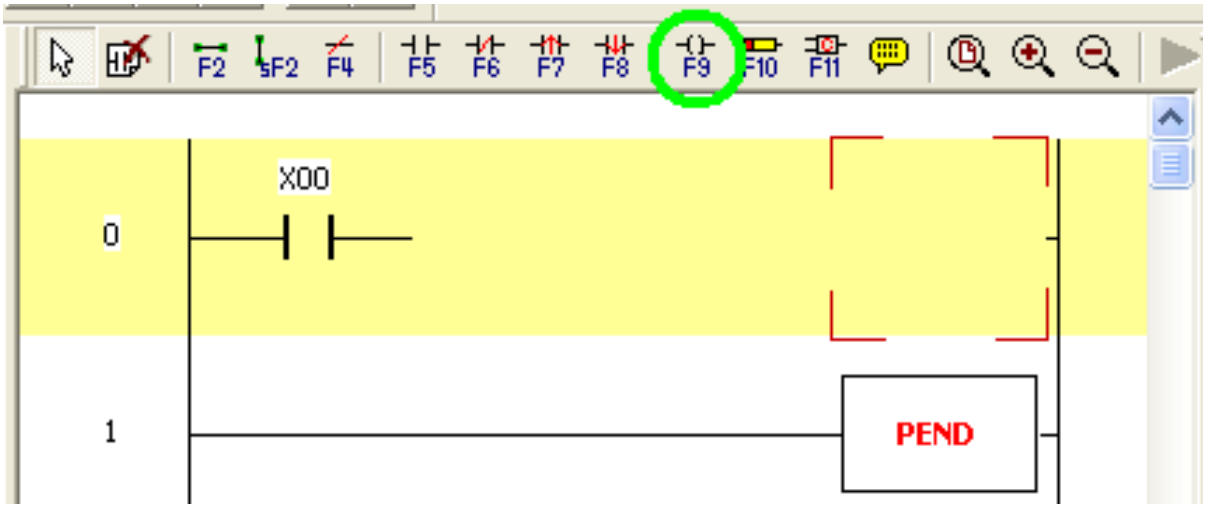


Figura 13: Icono marcado para contactos de salida.

Figura 14: ventana de inserción de dirección de contacto de salida

Una vez insertado el contacto de salida la disposición de elementos se tendrá que mostrar tal como se ve en la figura siguiente:

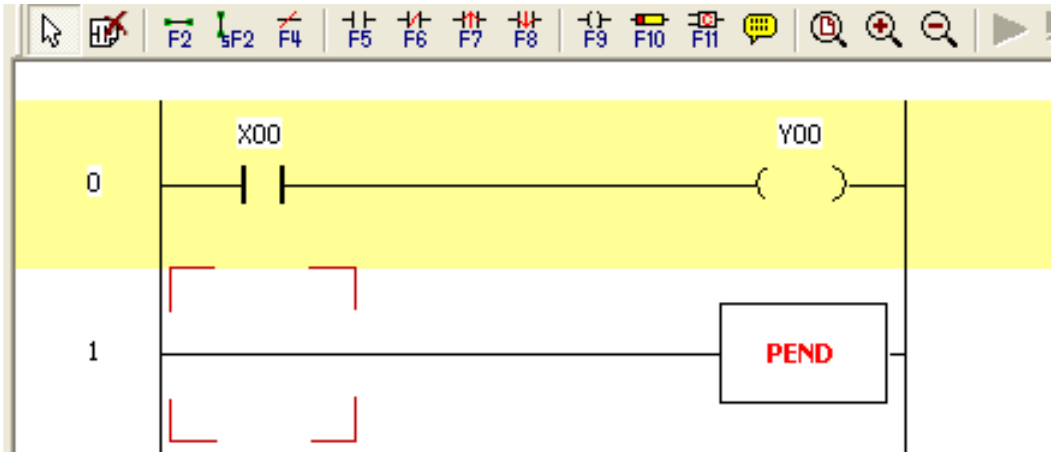


Figura 15: Distribución de elementos luego de insertar un contacto de entrada y de salida

Paso 6: ahora insertaremos un contacto normalmente cerrado para ello ubicamos el cursor del editor ladder (cuadro en rojo) en la posición final del programa, tal como se muestra en la figura 15.

Paso 7: luego damos clic en el icono sobre la barra de herramientas del editor ladder (ver figura 10) para agregar un contacto de entrada normalmente cerrado.

Paso 8: Adicionalmente aparecerá una nueva ventana solicitando la dirección del contacto de entrada a ingresar, para este caso ingresaremos la dirección X01, y damos clic en OK.

Paso 9: Luego ingresaremos un nuevo contacto de salida, para ello ubicamos el cursor sobre del editor ladder en el área de bobinas de salida, (lado derecho del contacto normalmente cerrado insertado en el paso anterior).

Paso 10: Dando clic en el icono sobre la barra de herramientas del editor ladder ingresaremos un nuevo contacto de salida, al cual daremos por dirección Y01.

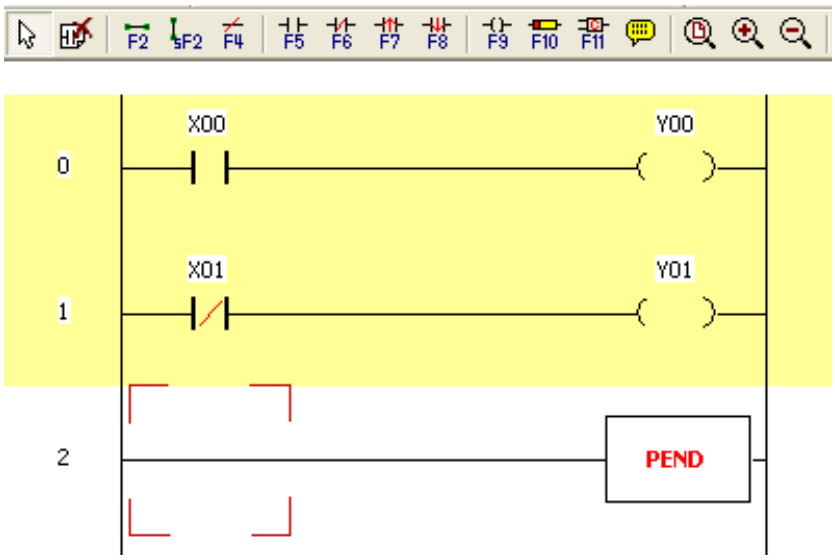


Figura 16: Editor ladder con contactos NA y NC

- Para ingresar rápidamente un contacto de entrada normalmente abierto, “NA” podemos utilizar solamente el teclado, presionando la tecla F5.
- Para ingresar rápidamente un contacto de entrada normalmente cerrado, “NC” podemos usar el teclado presionando la tecla F6.
- Para ingresar un contacto de salida de manera directa podemos simplemente presionar la tecla F9 del teclado.
- Antes de utilizar el teclado para la inserción de contactos es necesario ubicar correctamente el cursor del editor Ladder sobre el área donde se insertara dicho contacto requerido.

Luego de haber seguido los pasos anteriores correctamente se mostrara la distribución de los elementos tal como se muestra en la figura 16.

Paso 11: ahora ingresaremos 2 contactos de entrada NC y NO en serie, para ello dirigimos el cursor del editor ladder, a la última línea de programa.

Paso 12: Insertamos un contacto de entrada F5 y damos por dirección X02, luego insertamos otro contacto esta vez NC, con dirección de memoria X02, al lado derecho del contacto X02 previamente insertado, de tal manera que se encuentren en serie en una sola nueva línea de programa (ver figura 17).

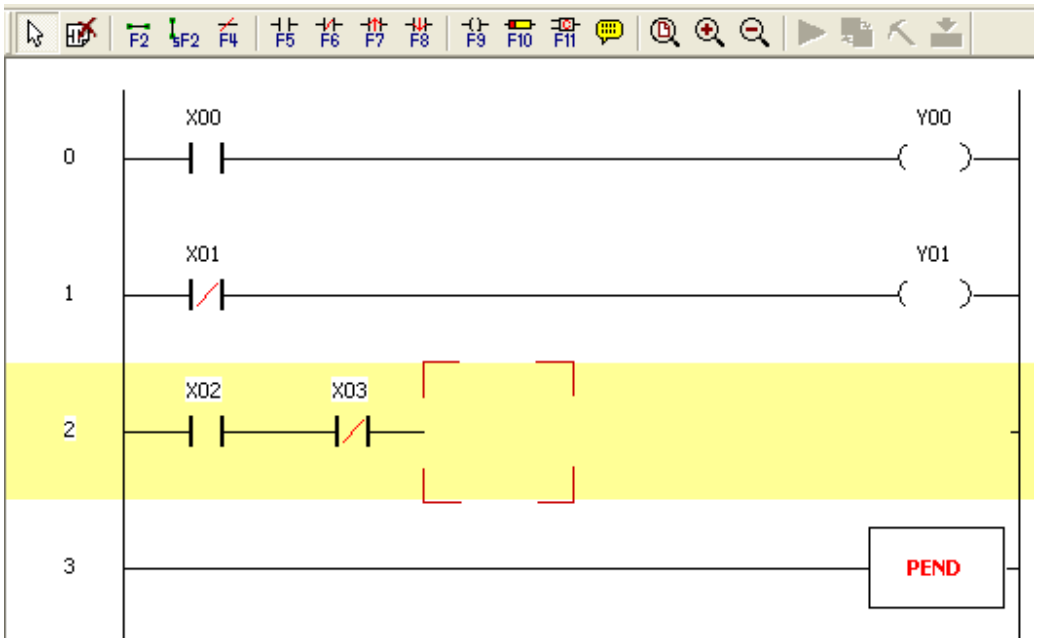


Figura 17: inserción de contactos de entrada en serie

Paso 13: Una vez dispuestos los contactos X03 y X04 en serie colocaremos a lado derecho de ambos contactos una contacto de salida con dirección Y02 (revisar paso 5 para mayores detalles de inserción de contactos de salida).

Paso 14: En los siguientes pasos diseñaremos una nueva rama con dos contactos en paralelo, para ello nuevamente trasladamos el cursor del editor a la última línea de programa.

Paso 15: Damos clic derecho sobre esta línea, que mostrara inmediatamente una lista de opciones a realizar (ver figura 18), de las cuales seleccionamos *"insert Line"* para insertar una nueva línea de programa.

Paso 16: Nuevamente trasladamos el cursor del editor a la última línea de programa e insertamos una nueva segunda línea de programa, tal como se describió en el paso 15, de tal forma que aparezcan dos líneas en blanco antes del bloque final **PEND** marcado en rojo.

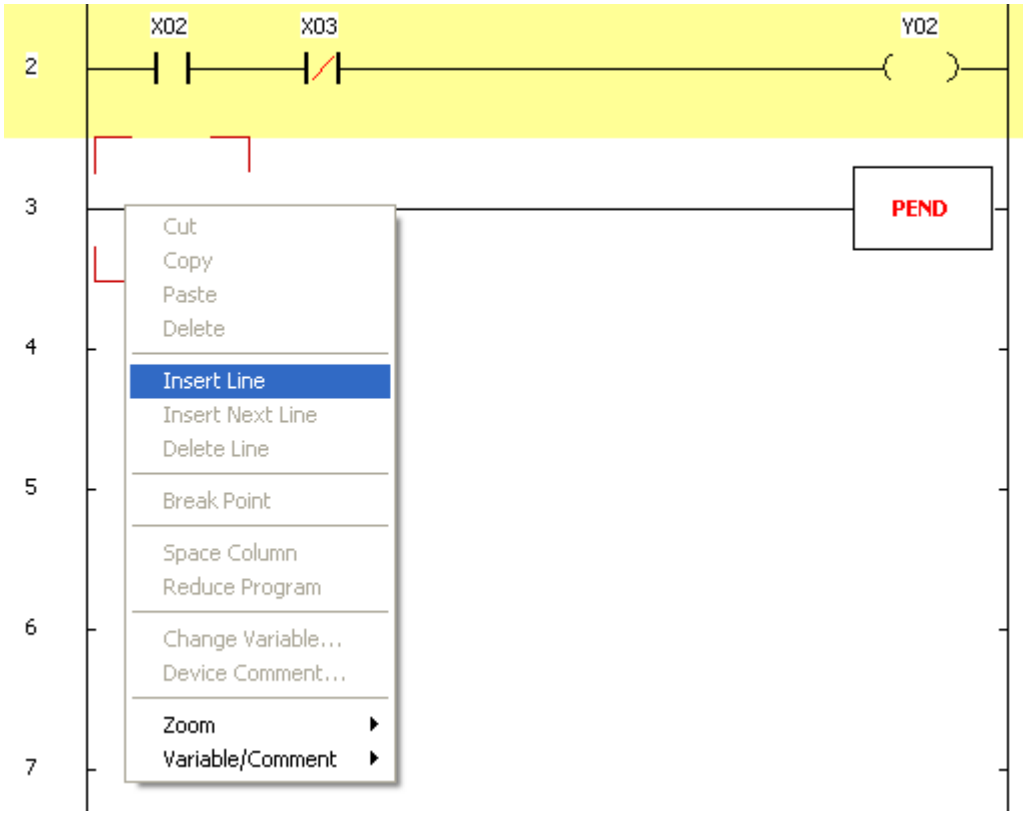


Figura 18: inserción de línea de programa en el editor Ladder

Paso 17: luego en la primera nueva línea creada insertaremos un contacto de entrada NO con dirección X04 y en la segunda línea inferior nueva otro contacto NC con dirección X05 tal como se muestra en la figura 19.

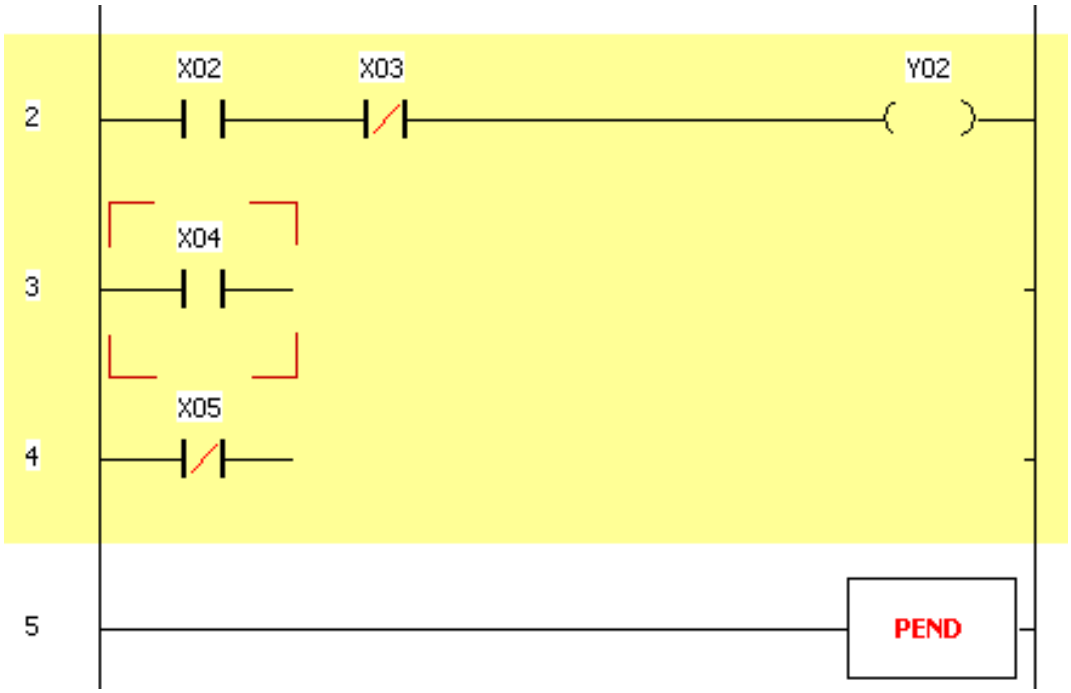


Figura 19: disposición de contactos de entrada para red en paralelo

Paso 18: Ahora ubicamos el cursor del editor ladder sobre el contacto X04 (ver figura 19) y damos clic en el icono de la barra del editor, para crear una rama vertical que conecte los contactos X04 y X05 en forma paralela.

Paso 19: una vez realizado el paso anterior podemos agregar un contacto de salida sobre el lado derecho del contacto X04, que represente el estado de la evaluación de las entradas en paralelo X04 y X05. Para agregar este contacto de salida ubicamos el cursor en el área contactos de salida y le damos la dirección de memoria Y03. (Ver figura 20).

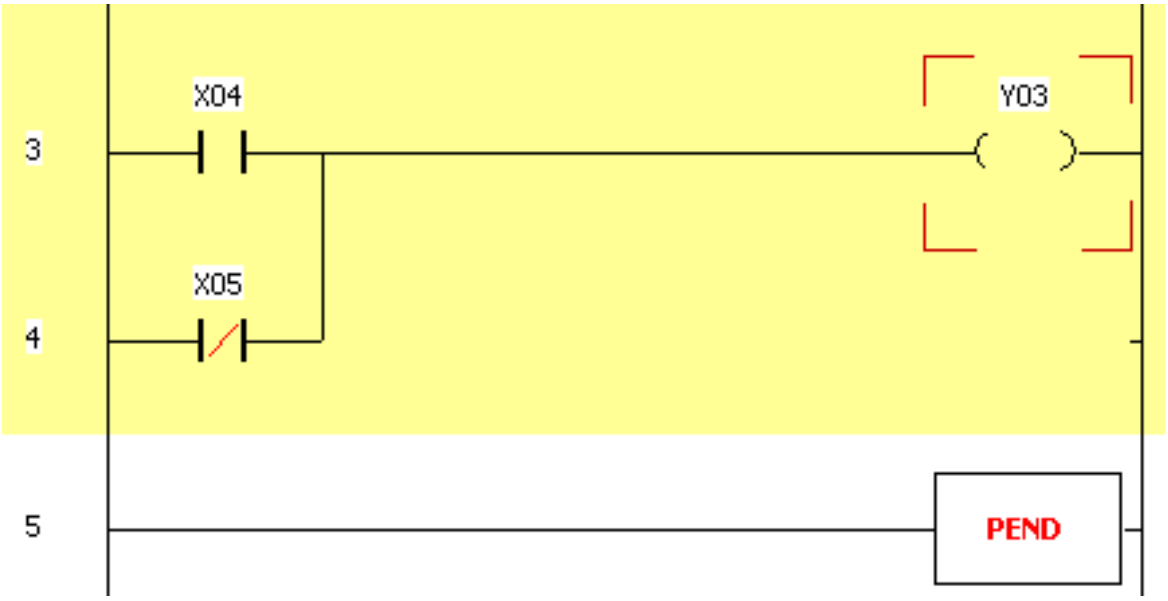


Figura 20: línea de programa con contactos de entrada en paralelo.

- Si en caso hubo errores en la disposición de los elementos del programa, que deseemos corregir o eliminar podemos hacer doble clic en cada elemento y editar su dirección de memoria, o suprimir dicho contacto con la tecla “Supr” del teclado.

Paso 20: Como final de ejemplo de programa a realizar, armaremos una red con combinación de contactos en paralelo y serie, para ello nuevamente trasladamos el cursor del editor a la última línea de programa.

Paso 21: Agregamos un contacto de entrada X06 “NA” en la primera columna de la última línea de programa, luego a su derecha otro contacto de entrada X07 “NA” y a la derecha de este otro contacto X08 pero en este caso “NC”, para finalizar la edición de la línea actual agregaremos un contacto de salida Y04, Ver figura 21.

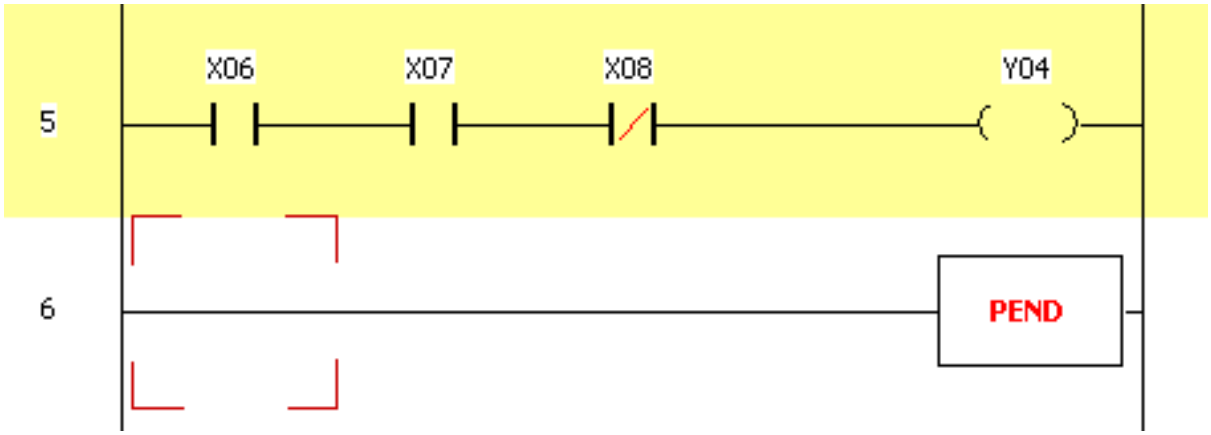


Figura 21: disposición de 3 contactos de entrada en serie.

Paso 22: Ahora crearemos una nueva fila de programa tal como se hizo en el paso 15.

Paso 23: Esta vez agregaremos un contacto de entrada X09 “NA”, debajo del contacto X07 creado anteriormente, ubicando antes el cursor del editor en la posición correspondiente, de tal forma que la disposición de los elementos queden mostrados de la siguiente forma:

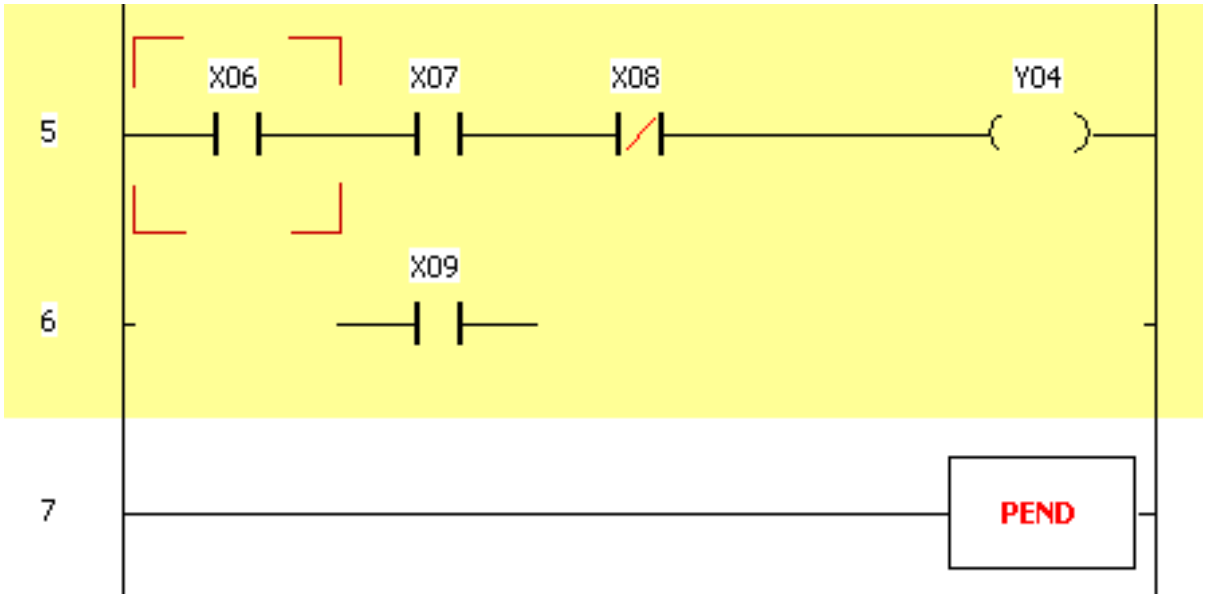


Figura 22: Contacto de entrada creado sin conexiones directas.

Paso 24: Tal como se ve en la figura 22, el contacto nuevo X09, no se encuentra conectado en alguna rama, para hacer la conexión en paralelo a la rama superior a esta ubicamos el cursor sobre el contacto X06, tal como se muestra en la figura 22, y hacemos clic en el icono de la barra de herramientas del editor, para crear una rama vertical que conecte el contacto X06 hacia X09.

Paso 25: Una vez realizado el procedimiento anterior colocamos el cursor del editor ladder justo al lado derecho del contacto X09, y hacemos clic en el icono F2 de la barra de herramientas del editor para crear una línea horizontal que permita unir el otro punto del contacto X09. El resultado de las instrucciones indicadas deberá mostrar la disposición de los elementos tal como se muestra en la figura 23.

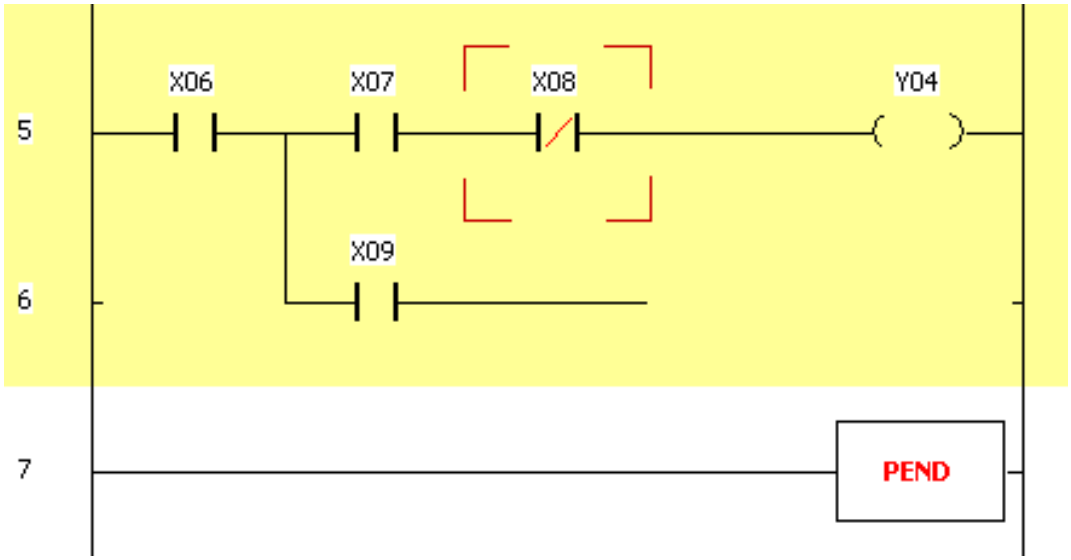


Figura 23: disposición de elementos para red paralelo serie.

Paso 26: Ahora ubicamos el cursor del editor ladder sobre el contacto X08, tal como se muestra en la figura 23, y damos clic en el icono SF2 de la barra de herramientas del editor, para unir la red X07-X08, en paralelo con el contacto X09. El resultado final de esta rama se mostrar de la siguiente manera:

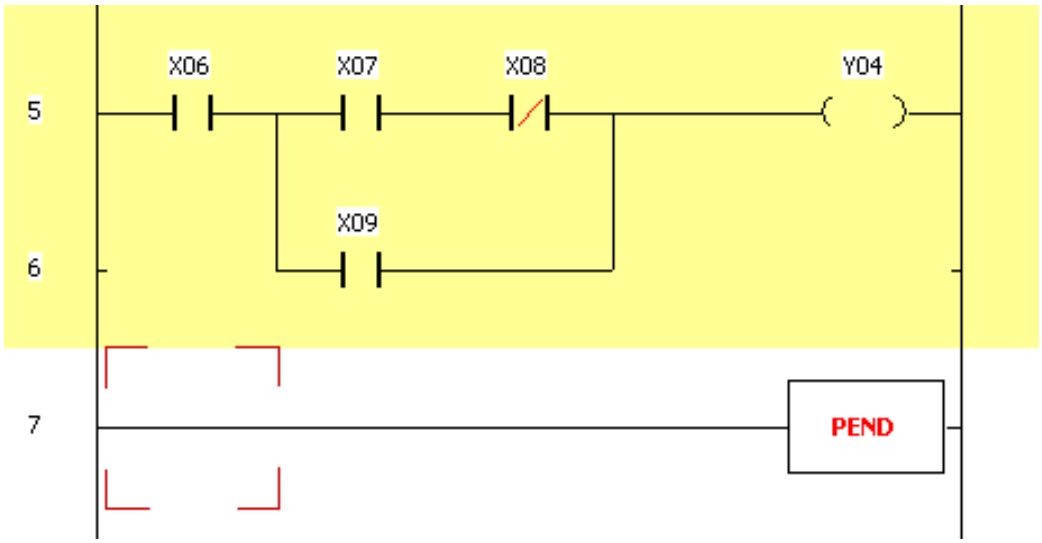


Figura 24: Red en serie paralelo de contactos de entrada

Paso 27: Para finalizar la edición de nuestro primer programa ejemplo ubicamos el cursor del editor sobre la última línea de programa (ver figura 24), y damos clic en el icono F10 para agregar la instrucción fin de programa "END". Al dar clic en el icono descrito aparecerá una pequeña ventana (Mostrada en la figura 25), en el cual escribiremos el texto "end" en la barra de ingreso de texto, para luego dar clic en el botón OK.

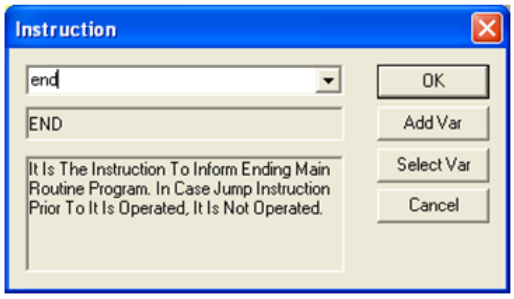

Instruction
end
END
It Is The Instruction To Inform Ending Main Routine Program. In Case Jump Instruction Prior To It Is Operated, It Is Not Operated.
OK
Add Var
Select Var
Cancel

Figura 25: ventana de inserción de instrucciones.

Una vez terminado los pasos correctamente el programa creado en el editor ladder se mostrara de la siguiente manera.

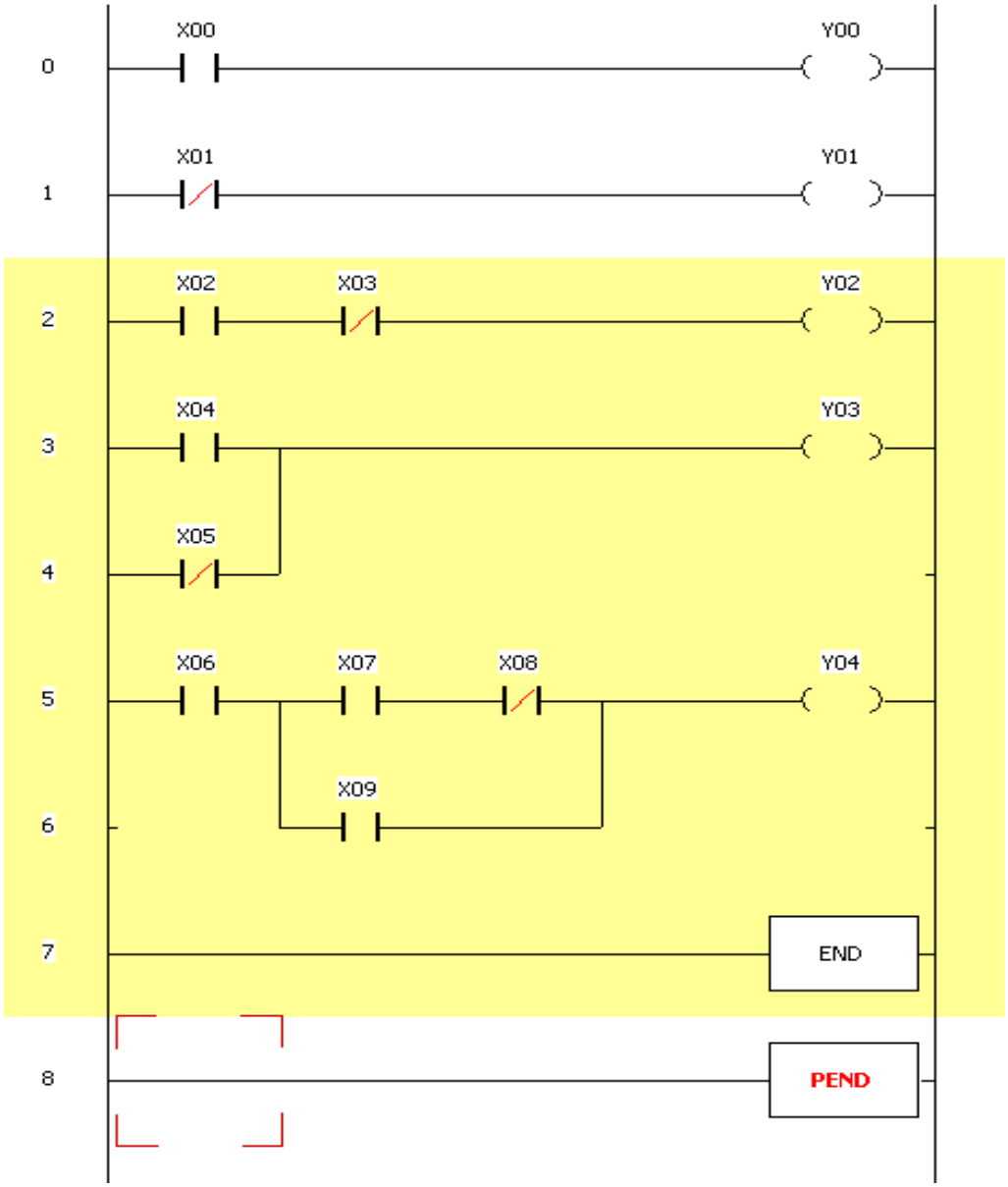


Figura 26: Diseño final del programa basico ejemplo

- El área que aparece sombreada de amarrillo sobre el editor ladder solo representa las últimas modificaciones realizadas sobre el programa actual.
- Si en caso hubo errores en la disposición de los elementos del programa, que deseemos corregir o eliminar podemos hacer doble clic en cada elemento y editar su dirección de memoria, o suprimir dicho contacto con la tecla "Supr" del teclado.
- Al haber seleccionado un programa del Tipo ***"Scan Program"***, este bloque de programa se estará evaluando constantemente desde la línea 0 hasta la línea donde se encuentre la instrucción de finalización (Bloque *END*).
- Para agregar cualquier otra instrucción siempre utilizamos el icono

de la barra de herramientas del editor, y transcribimos el nombre y parámetros de la instrucción sobre la ventana de inserción de instrucciones (ver figura 25). Para mayor información del manejo de instrucciones ver el tema 7 de la presente guía.

Una vez terminado este programa básico, este proyecto ya puede ser compilado, y transferido un Controlador PLC, conectado sobre algún puerto de la computadora.

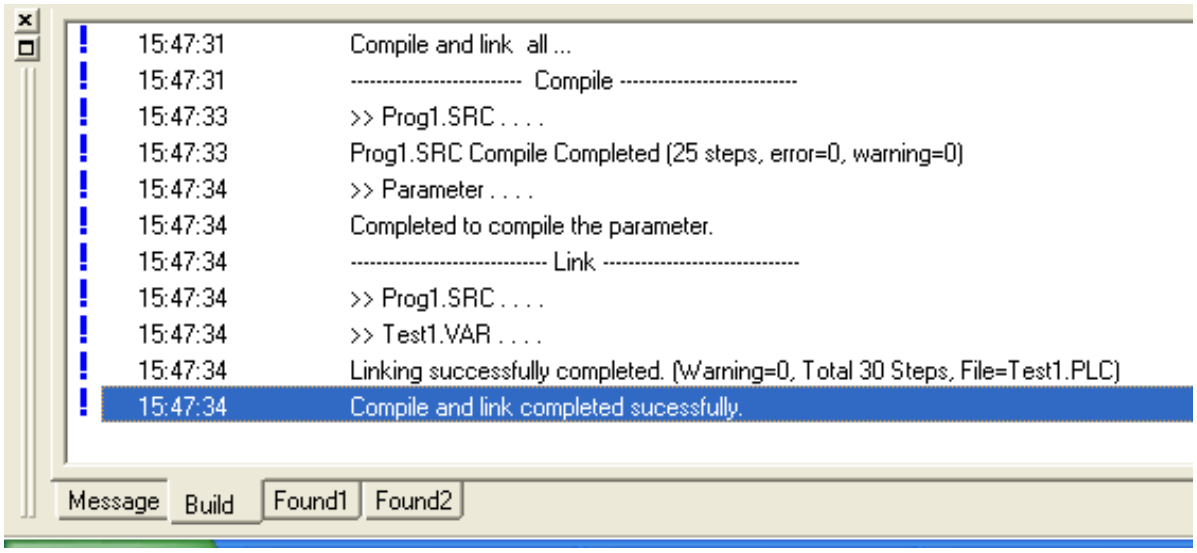
4. COMPILACIÓN Y DESCARGA DE PROGRAMA HACIA PLC

Una vez creado un programa, podemos verificar si la programación realizada es la correcta, efectuando las herramientas de compilación

4.1 Compilación de programa

Para acceder directamente a las opciones de compilación nos dirigimos a la barra de menú principal y herramientas, >Tool / Compile All+Link
Automáticamente el software CICON, evaluara todas las configuraciones realizadas, y generara los archivos de control a descargar sobre el PLC.

Para verificar el estado del resultado de la compilación, en la ventana de mensajes se mostrara el registro de sucesos, tal como se muestra en la siguiente figura:


15:47:31 Compile and link all ...
15:47:31 ---------- Compile ----------
15:47:33 >> Prog1.SRC . . . .
15:47:33 Prog1.SRC Compile Completed (25 steps, error=0, warning=0)
15:47:34 >> Parameter . . . .
15:47:34 Completed to compile the parameter.
15:47:34 ---------- Link ----------
15:47:34 >> Prog1.SRC . . . .
15:47:34 >> Test1.VAR . . . .
15:47:34 Linking successfully completed. (Warning=0, Total 30 Steps, File=Test1.PLC)
15:47:34 Compile and link completed sucessfully.
Message
Build
Found1
Found2
Figura 27: ventana de mensajes con resultados de mensajes

4.2 Descarga de programa hacia PLC

Para iniciar la descarga es necesario seleccionar el medio de comunicación entre la PC y el PLC, existen 3 opciones principales de comunicación, USB, Serial RS232, Ethernet.

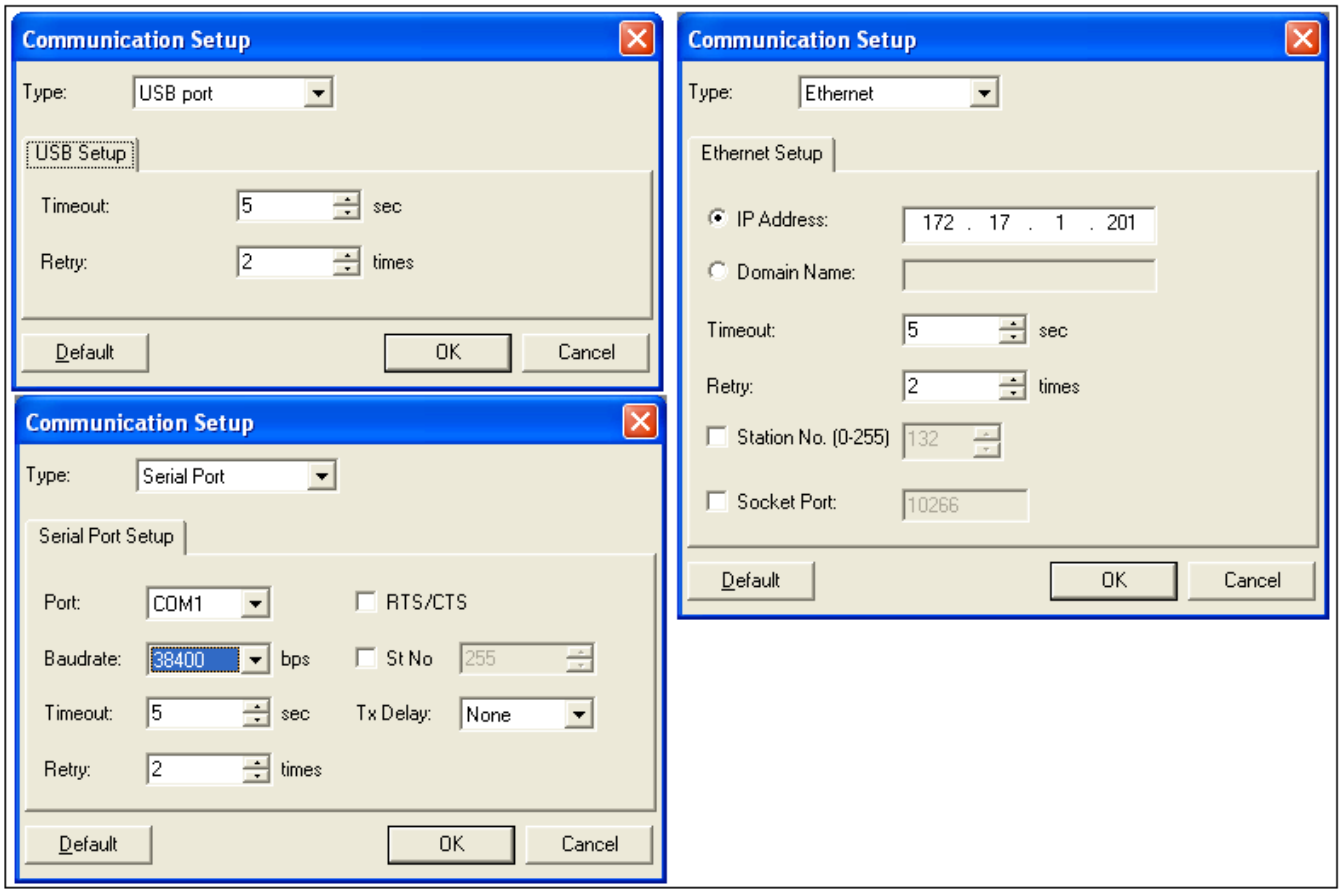


Figura 28: ventana de opciones de comunicación via USB – Serial - Ethernet

La comunicación más básica que no requiere configuraciones en el PLC, es vía USB, en este caso solo es necesario tener instalado los drivers de comunicación vía el puerto USB de la PC.

Los drives de comunicación vía USB se instalan automáticamente con el software CICON cuando se utiliza Windows XP, para el caso de Windows 7 es necesario seguir los pasos indicados en el manual de “instalación de drivers USB para Windows 7”.

4.2.1 Procedimiento para descargar programa del PLC vía el puerto USB

Para establecer la conexión física entre la PC y el PLC usamos el cable conector de Mini USB a USB. Luego sobre el software CICON Seleccionamos las opciones de comunicación vía usb sobre el menú principal >*Tool / Connection Setup…* tal como se indica en la siguiente figura:

Seleccionada la opción correspondiente es necesario establecer la conexión con el PLC, esto se realiza a través del menú principal >*Online / Connect*

Cuando la conexión se ha establecido en la ventana de mensajes se podrá observar el estado de éxito de la comunicación en línea a si mismo sobre el área de árbol de proyecto se podrá actualizar la lista de módulos conectados sobre el PLC.

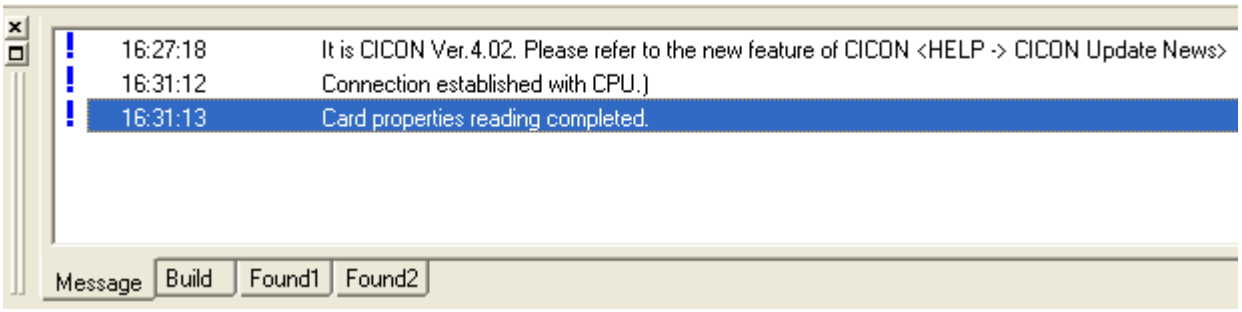

16:27:18 It is CICON Ver.4.02. Please refer to the new feature of CICON <HELP -> CICON Update News>
16:31:12 Connection established with CPU.)
16:31:13 Card properties reading completed.
Message
Build
Found1
Found2

Figura 29: Estado de la conexión en línea entre la PC y el PLC

Una vez que el estado de conexión es exitoso recién es posible descargar dicha aplicación sobre el PLC. Para ello es necesario dirigirse al menú principal >*Online / Download…*

Ejecutado dicho comando se efectúa un proceso de compilación adicional, luego de confirmado que no existan errores, se procede con la descarga, durante este procedimiento aparecerán diversas ventanas de dialogo para prevenir la descarga no deseada sobre el PLC, y otras opciones.

La primera ventana que aparece es la que se muestra en la figura 30, el cual pide la confirmación de la detección en la ejecución del programa del PLC, si es que antes el controlador ejecutaba su aplicación previamente programada.

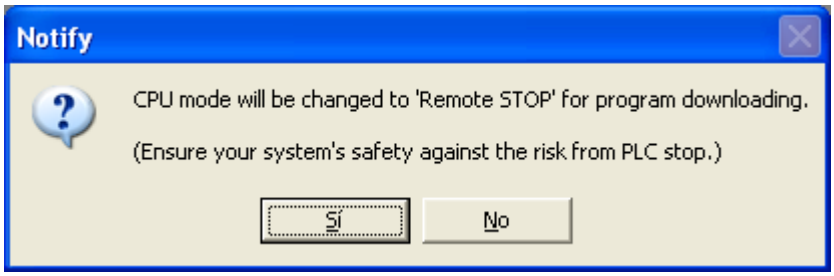

Notify
CPU mode will be changed to 'Remote STOP' for program downloading.
(Ensure your system's safety against the risk from PLC stop.)
Sí
No

Figura 30: orden de parada de PLC

Luego de confirmar la opción de “Stop”, aparecerá otra ventana para confirmar la descarga del programa.

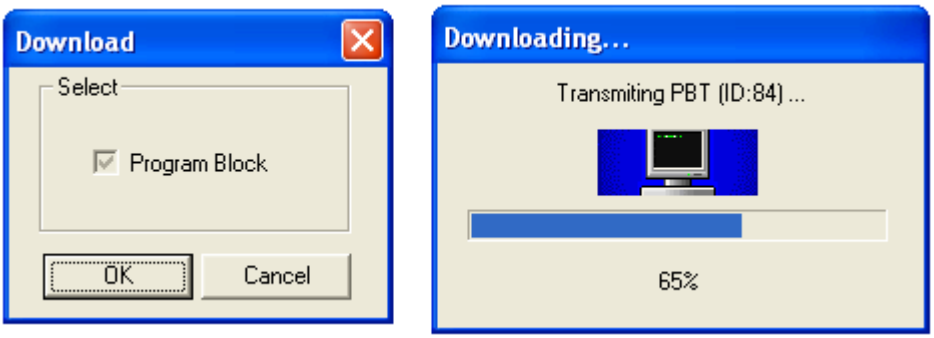


Figura 31: ventanas de confirmación y descarga de Programa.

Al finalizar la descarga del programa aparecerá otra ventana que solicitara ejecutar la aplicación descarga sobre el PLC, Modo Run.

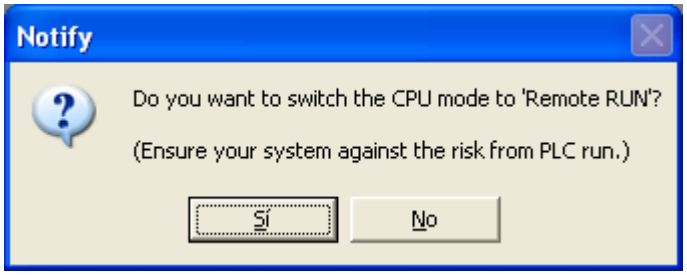


Figura 32: notificación a modo Run.

Seleccionando la opción “Si” comenzaremos a ejecutar la aplicación descargada sobre el PLC.

Para confirmar que el proceso de descarga se ha realizado con éxito debemos visualizar en la ventana de mensajes “Download Succeded”.

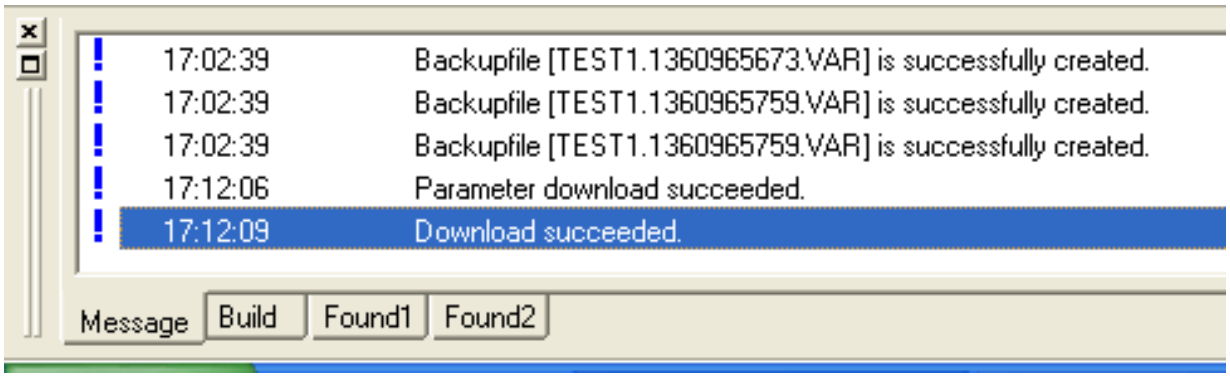


Figura 33: ventana de mensajes luego de descarga de programa

Si en caso no se confirmase el mensaje de descarga exitoso, es necesario realizar otro procedimiento de descarga.

4.2.1 Conexión del PLC vía el puerto Serial

Para este caso es necesario seleccionar sobre el menú de conexión, (figura 28), la opción Type: (Serial Port). Luego establecemos la conexión física con la computadora mediante el esquema mostrado en la figura 91.

Para la comunicación serial es necesario especificar el puerto serial a utilizar además de la velocidad de Baudios al cual se establecerá la comunicación entre la PC y el PLC, por defecto se encuentra 38400, sin embargo este valor puede ser alterado según la última configuración realizada sobre el PLC.

Cuando no se puede establecer la comunicación entre el PLC y la computadora debemos verificar las conexiones realizadas a si como verificar que otro software no esté usando el puerto serial de la PC direccionado por el software CICON.

Después seguimos los pasos anteriores mostradas en la conexión via USB para la descarga del programa.

4.2.1 Conexión del PLC vía el puerto Ethernet

Solo es posible descargar un programa u aplicación al PLC via red Ethernet si se conoce su dirección IP, configurada anteriormente via una interfaz USB u Serial.
Una vez que conocemos la IP del PLC, realizamos la conexión física utilizando un cable Ethernet a través de un Switch.

Luego en el menú de conexión, (figura 28), seleccionamos la opción Type: Ethernet y colocamos sobre los campos mostrados la dirección IP del PLC a establecer comunicación.

Para asignar antes una dirección IP al PLC es necesario registrar dichos valores en el menú de parámetros del PLC configurable en el software CICON, ver figura 64.

Para acceder a la configuración de parámetros del PLC, nos dirigimos al menú principal, *Tool / PLC Parameter..*

Una vez que hemos configurado estos parámetros, es necesario realizar una descarga del programa y configuraciones hacia el PLC.

- Es importante destacar que cuando se realiza una carga de programa al PLC, toda la lista de parámetros sobre el menú anteriormente descrito es actualizado y cargado hacia el PLC, por eso cuando se quiera realizar una comunicación via Ethernet, es necesario verificar que la IP dentro del menú de parámetros sea la que se requiera para la aplicación.

5 COMUNICACIÓN EN LÍNEA CON EL PLC

La conexión en línea con el PLC se realiza con el software CICON, de una manera sencilla, asi mismo el reconocimiento del hardware interconectado es automatico. Siempre para realizar una conexión en línea se debe hacer clic en el menú principal > *Online / Connect*

5.1 Conexión en línea en modo monitor

Este modo le permite visualizar el estado de los contactos y bloques del diagrama ladder de modo que se pueda ver el valor de las variables como el estado de las salidas y entradas dispuesta sobre el diagrama de programación.

Para seleccionar esta opción incluyendo la descarga reiterativa del programa es necesario dirigirse al menú principal > Online / Link + Download +Monitor …

Para seleccionar solamente la opción de Monitor, se debe ubicar el icono control sobre la barra de herramientas del editor (Ladder) y presionar el botón para activar y desactivar la opción monitor.

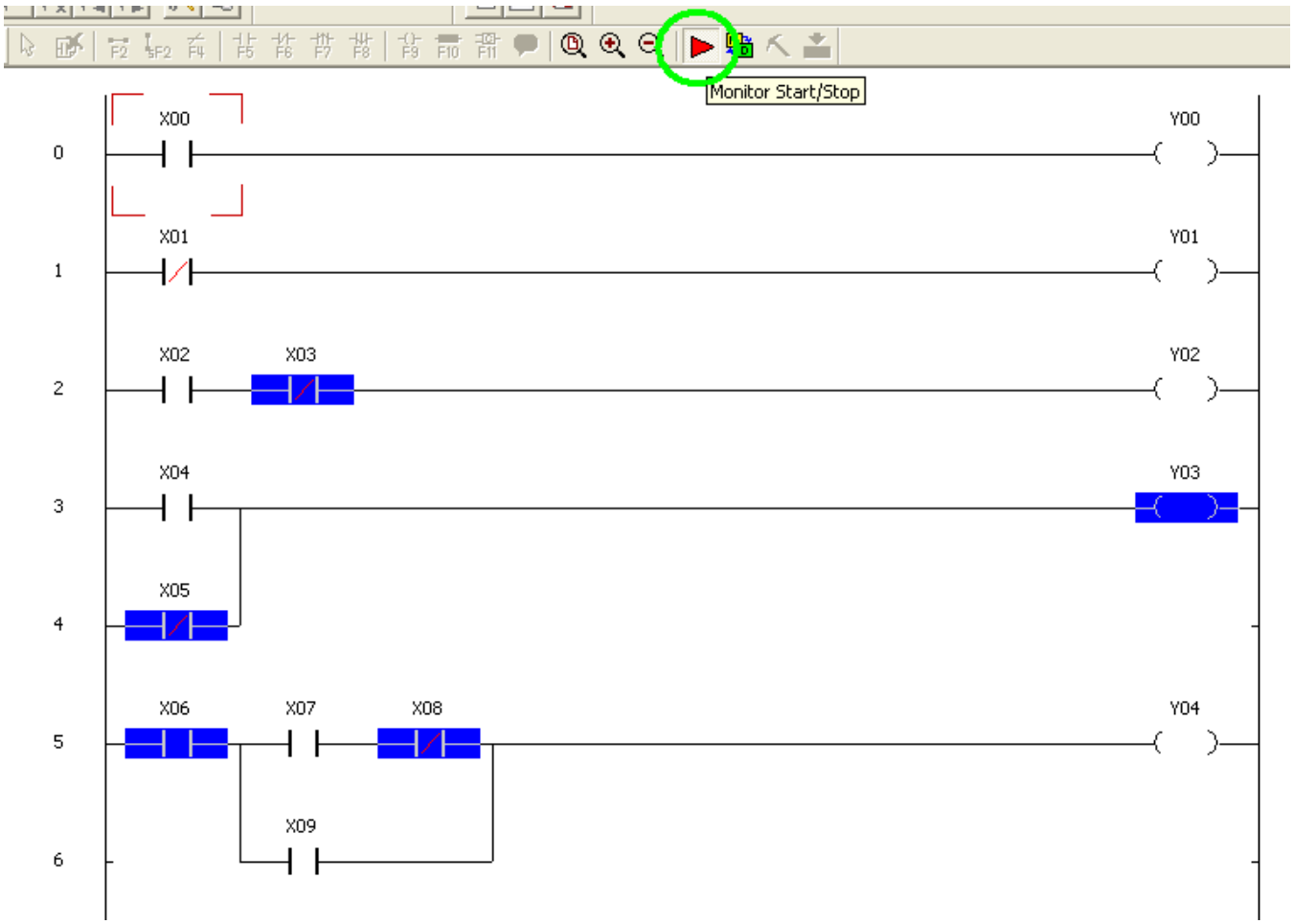


Figura 34: selección de opción monitor

5.2 Estado de operación de PLC

para poder evaluar el estado del PLC y obtener un diagnostico general de la aplicación programada, tenemos primero que establecer una conexión en línea: menú principal > *Online / Connect*

Luego sobre el menú principal > *Online / PLC Status…*
Aparecerá la ventana siguiente con los datos relativos al PLC en conexión:

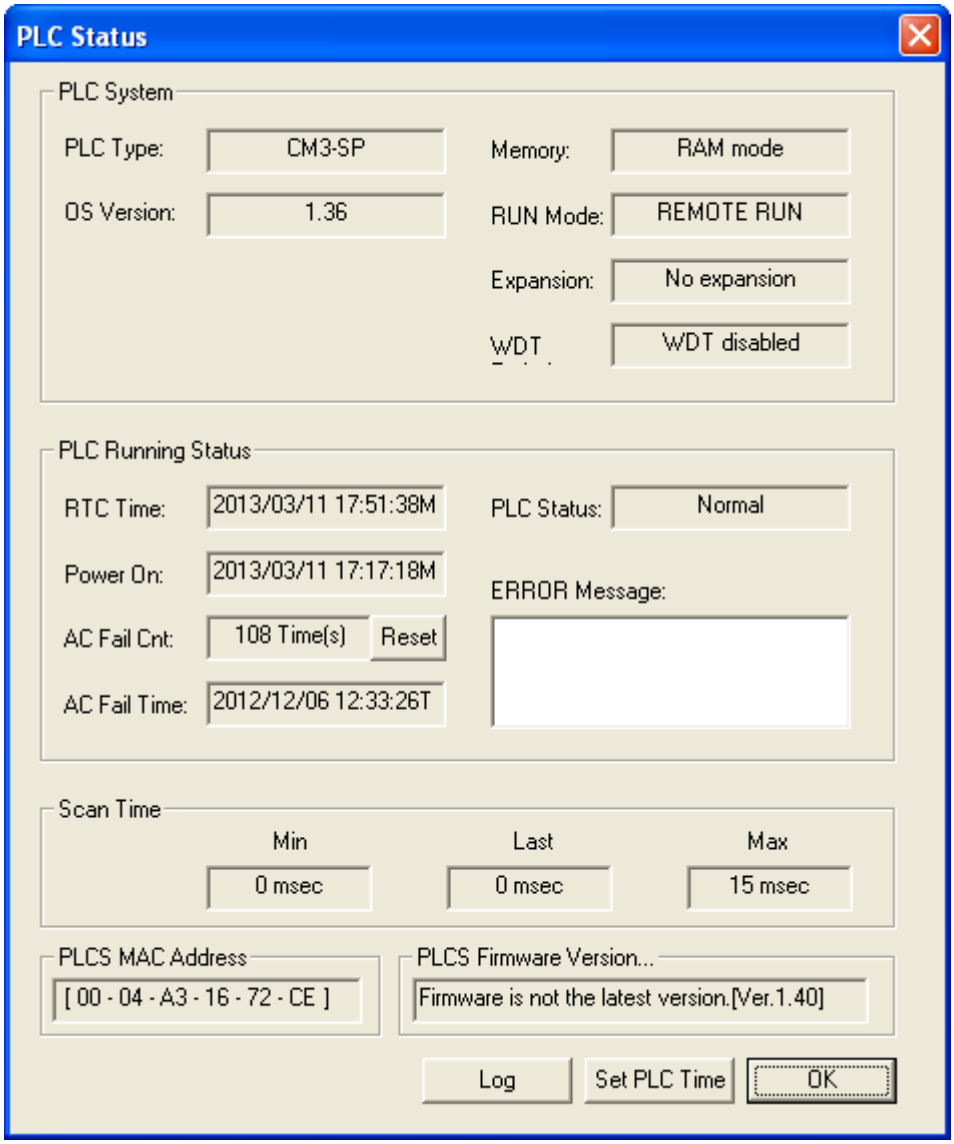


Figura 35: ventana de estado de PLC – on line

En esta ventana podremos observar el estado de ejecucion del PLC, version de firmware, reloj del sistema, mensajes de errores, tiempo de escaneo de programa, dirreccion Mac del dispositivo.

Si presionamos el boton Log, podremos vizualizar el registro de eventos almacenados sobre el PLC

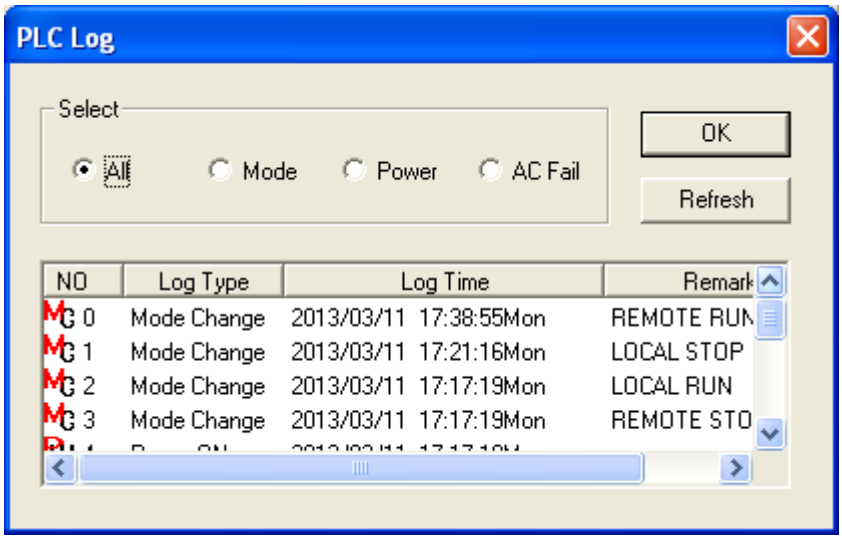


Figura 36: Ventana de registro de eventos

6. TIPOS DE VARIABLE Y MEMORIA INTERNA

El PLC cimon posee 11 tipos de variable almacenadas en su memoria interna, cada tipo de variable se designa dentro del entorno de programacion de CICON, como “Device” y su nomenclatura difieren según su tipo.

6.1 Lista de variables de memoria

La lista de variables se pueden clasificar en 3 tipos, tipo bit, tipo entero y tipo cadena.

X (input):
Esta variable es utilizada para dirreccionar un entrada digital fisica del PLC, o variables de entrada sobre modulos I/O.

Y (output):
Esta variable es utilizada para dirreccionar un salida digital fisica del PLC, o variables de salida sobre modulos I/O.

M , L (I/O):
Esta variable es de uso general para almacenamiento temporal dentro de la aplicación, puede ser dirreccionada tanto como bit o entero de lectura/escritura.

K (Latch Relay):
Esta variable es de uso general para almacenamiento temporal dentro de la aplicación, con caracteristicas similares a las variables M y L, la diferencia es que este tipo de variable puede almacenar su ultimo estado logico cuando el suministro de energia es quitado, no es necesario configurar parte del area de memoria con retencion.

F (Flag):
Este tipo de variable es de solo lectura, provee varias funciones utiles ya implementadas que pueden ser tomadas como referencia de estado, o temporizaciones dentro del PLC.

T (Timer):
Este tipo de variable es usada en los temporizadores, difiere de los otros tipos de variable porque puede dirreccionar al mismo tiempo variables tipo bit y tipo entero. Esto esta determinado según el tipo de funcion o instruccion al cual se le ha asignado.

C (Counter):
este tipo de variable es usado en variables del tipo contador su flexibilidad en su uso es igual que la variable tipo Timer, ya que puede dirreccionarme como variable tipo bit y tipo entero escribiendo la misma nomenclatura. Se define según la funcion o instrucción al cual se le ha asignado.

S (Controlador de paso):
Este tipo de variable se utiliza para el control del algoritmo en la mayoria aplicaciones no es necesario utilizarlo.

D (Word data):
Este tipo de variable es de proposito general, es muy usado para el almacenamiento de datos del tipo entero, doble entero y float.

Z (Call stack):
Almacena el valor temporal de la pila de memoria que se genera cuando se realiza llamadas de subrutina, es una variable del tipo word, la capacidad de memoria de esta variable determina el numero maximo de saltos o niveles para llamadas de subrutina que se podran realizar sobre el proceso. Para los PLC cimon el nivel maximo de niveles es de 16.

6.2 Nomenclatura de variables en programa

las dirrecciones de memoria estan ordenadas por palabras de 16 bits, es decir cuando se requiera dirreccionar una variable tipo bit se tendra que utilizar sobre una palabra o registro hasta en 16 posibilidades.

figura 37: Dirreccionamiento de variables tipo bit y word en variable tipo Y

- Cuando se dirrecciona variables tipo bit, se tiene que tener en cuenta que se debe usar valores hexadecimales cuando se sobrepase el bit 9 de un registro.
- para variables tipo word, la nomenclatura siempre es decimal.

Puede obviarse el uso de escritura de varios ceros a la izquierda cuando se escribe las variables:

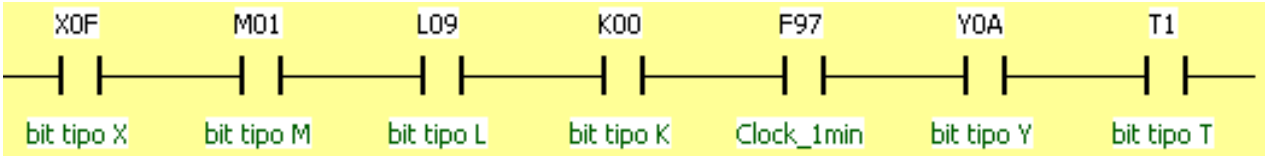


figura 37: nomenclatura de variables tipo bit

Para variables tipo word, siempre se deja un cero a la derecha cuando se utiliza las variables X,Y,M,K,L.

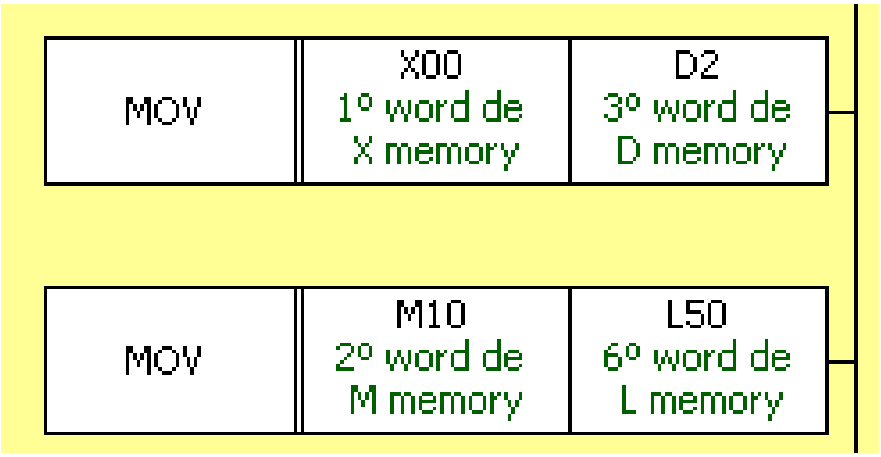


Figura 38: nomenclatura de variables tipo word

7. MANEJO DE INSTRUCCIONES BASICAS LADDER

Dentro del repertorio extenso de las instrucciones implementadas en el PLC Cimon, es importante primer conocer las instrucciones mas basicas y comunes utilizadas en los proyectos, a continuacion describiremos cada grupo de instrucciones.

7.1 Instrucciones tipo flanco de bit

Estas instrucciones se implementan dentro de un contacto de entrada, detectando el impulso positivo o negativo del estado de una variable tipo bit.

En el ejemplo siguiente se observa que cuando las instrucciones tipo contacto muestran una flecha hacia arriba o hacia abajo, estas representan el tipo de deteccion flanco que pueden realizar.

cuando ocurre un flanco positivo en M00, la salida Y11, se activa solo por un tiempo muy pequeño equivalente al tiempo instantaneo de scaneo, que es el tiempo en que el PLC ejecuta todo el programa por un ciclo.
De igual manera cuando ocurre un flanco negativo, a través de la seguna linea del ladder mostrada abajo, la salida Y12 permanece activa.

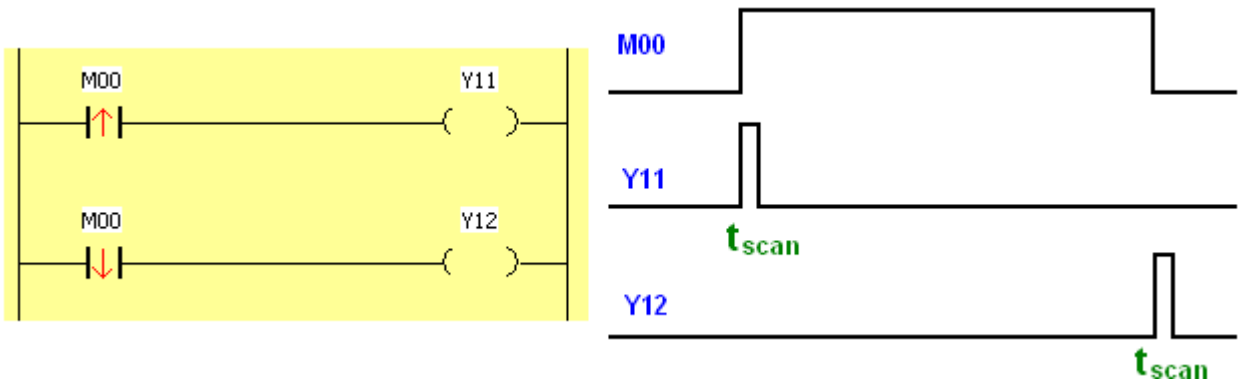


figura 39: operación de instrucciones de flanco.

7.2 Instrucciones tipo Bit Set / Reset

estas instrucciones permiten enganchar el estado de un contacto de salida a un valor establecido con tan solo un solo impulso de activacion, para dicho valor de salida ya sea a cualquiera de los valores binarios logicos conocidos 0 (reset) o 1 (set).

Estas funciones dentro de la programacion del PLC cimon, se llaman SET y RST, su nomenclatura se describe con el nombre de la instrucción seguida del bit de dirreccion, ejemplo: Set Y01

- solo puede aplicarse esta instrucción a contactos de salida modificables, es decir no puede dirreccionar las variables de entrada tipo bit, X.
- Estas instrucciones siempre requieren un contacto de entrada a la izquierda para su implementacion.

En el ejemplo siguiente se ve la aplicación de estas instrucciones, tambien se han agregado las instrucciones tipo flanco que agregan el filtrado de las entradas de control cuando estas presenten bajo el mismo estado logico y se requiera adicionar seguridad en la activacion de dicho contacto bajo las condiciones normales de operación.

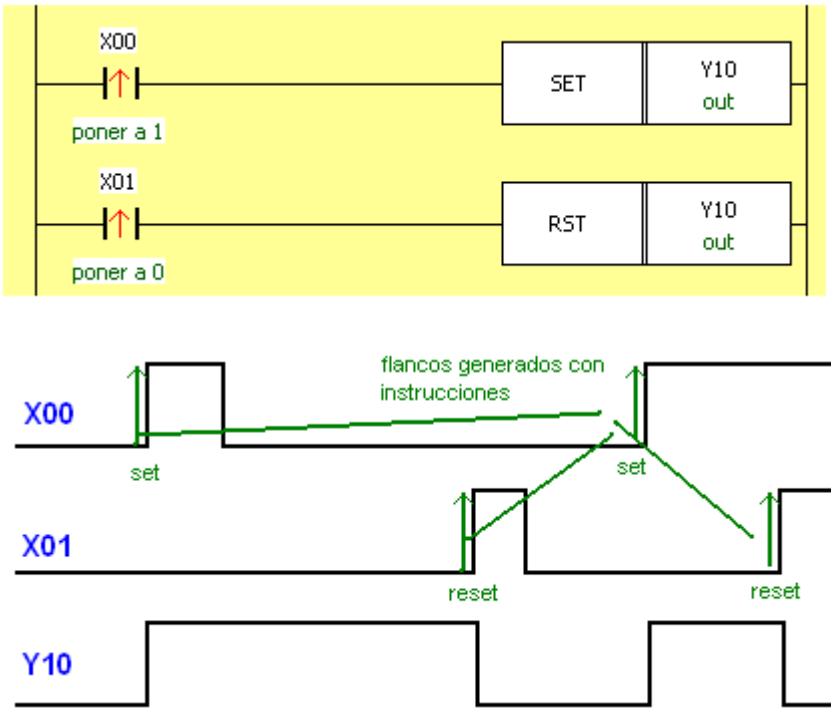


Figura 40: operación de las instrucciones SET y RST

En el grafico anterior se ve que si ocurre un flanco positivo en X00, la salida Y10 queda enganchada hasta que aparezca un nuevo flanco a través de X01.

7.3 Instrucciones de movimiento y copia de datos

Con estas instrucciones podemos copiar los datos desde un registro de origen a otro registro destino, este tipo de instrucción no puede utilizarse con variables tipo Bit.

Las instrucciones principales son MOV, DMOV, BMOV

con la instrucción MOV, solo podemos copiar datos de 16 bit (1word)
con la instrucción DMOV, es posible copiar datos de doble entero o variables tipo float de 32 bit (2word).

Con la instrucción BMOV, podemos copiar todo un bloque de memoria indicando el numero de registros de 16 bits que deseamos traspasar.

- Este tipo de instrucciones no puede ponerse individualmente en una misma linea de programa ladder, para ello siempre se requiere que se anteponga algun tipo de contacto a la izquierda.

La nomenclatura de estas estas instrucciones siempre comienza con el nombre de la funcion seguido siempre primero del registro origen y luego del destino.
Ejemplo: MOV D3 D4.

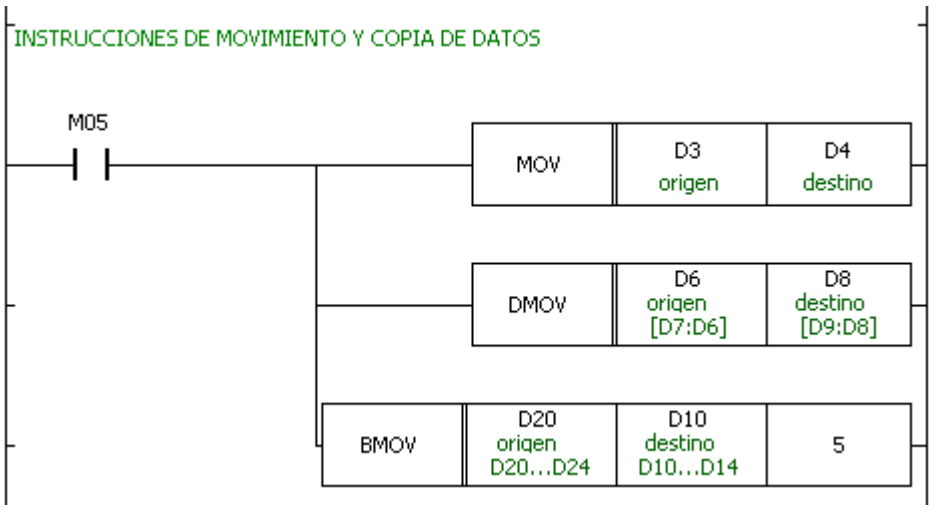


Figura 41: Instrucciones de movimiento

7.4 Instrucciones aritmeticas

Los PLC Cimon soportan variadas funciones matematicas que pueden operar a nivel de enteros, doble entero y numeros Float.

7.4.1 Instrucciones de suma y resta

Las funciones de suma y resta utilizan las instrucciones ADD y SUB, cuando se requiera sumar y restar registros de doble palabra siempre se agrega la letra D, ejemplo: DADD, DSUB.

Para operaciones con numeros flotantes, se antepone la letra E, ejemplo: EADD,ESUB

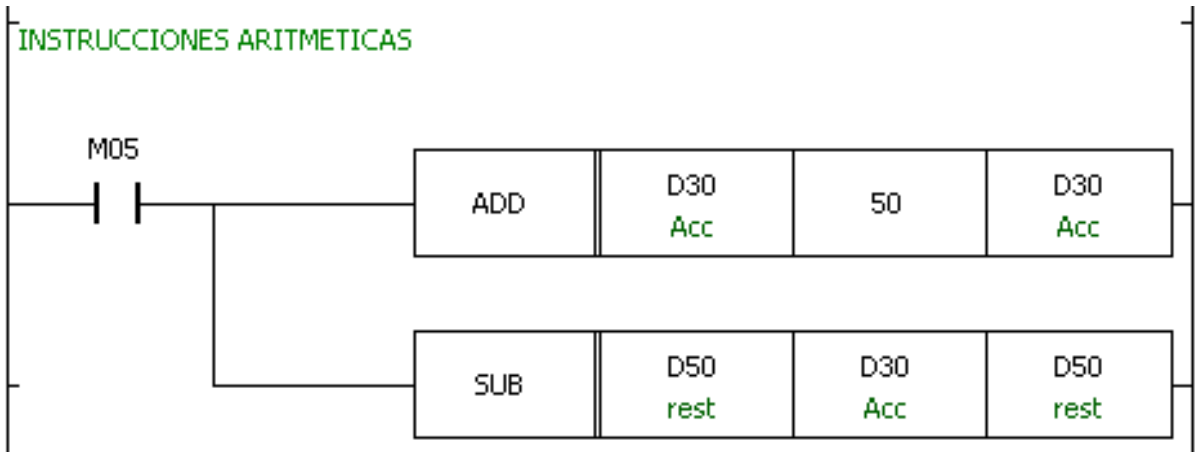


Figura 42: instrucciones de suma y resta.

En la figura anterior se muestra una instrucción de suma y otra de resta, el formato de escritura obedece al sistema de programacion, en el que el resultado o destino de la operación aparecera siempre en el lado izquierdo:

ADD D30 50 D30 :: D30 + 50 -> D30
SUB D50 D30 D50 :: D50 – D30 -> D50

Tambien existen las instrucciones basicas de incremento y decremento INC, DEC, estas funciones incrementan o decrementan en una unidad el valor de un registro asociado, ejemplo: INC D2, DEC D20

7.4.2 Instrucciones de multiplicacion y division

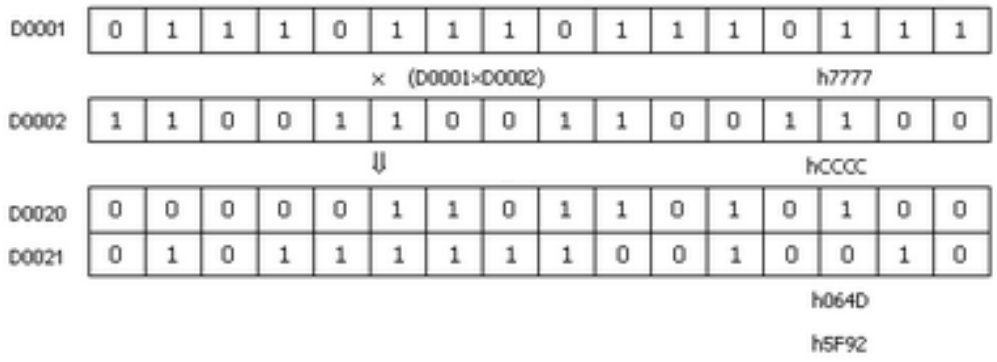
En este tipo de instrucciones tambien podemos alternar el tipo de variable a operar, entero, doble entero o numero flotante.

las instrucciones para enteros son MUL, DIV
las instrucciones para doble enteros son DMUL, DDIV

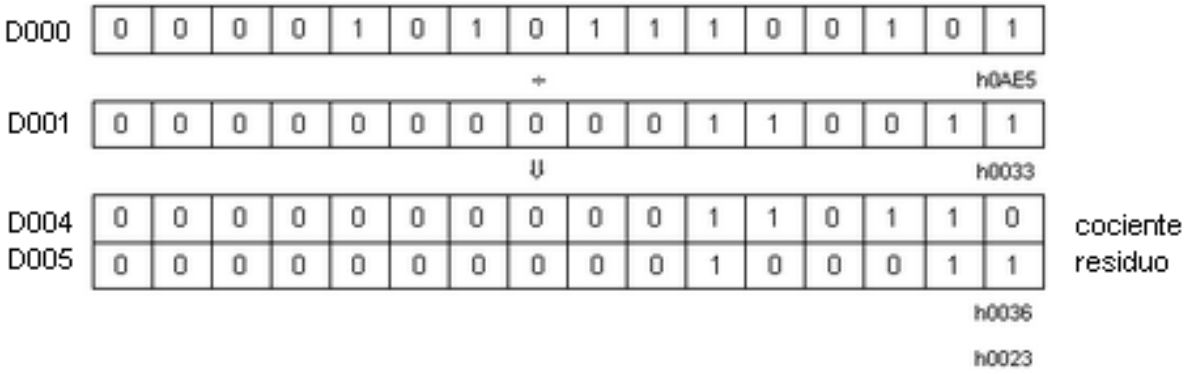
Cuando se utiliza la instrucción MUL, el resultado se almacena en dos registros continuos, asimismo cuando se utiliza la instrucción DIV, el residuo se almacena en uno de los 2 registros de resultado:

ejemplo:

MUL D0001 D0002 D0020



DIV D000 D001 D004

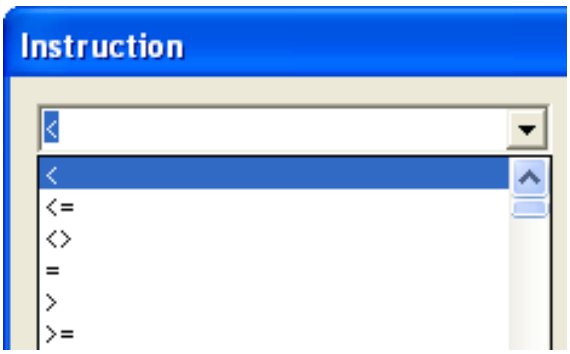


- cuando se agrega la letra W, al inicio de la instrucción se define que el resultado solo contenga un valor entero (16 bits). Ejemplo (WMUL, WDIV)
- cuando se agrega la letra DW el resultado siempre tendra el formato de doble palabra.
- Para operaciones con numeros Float, se tiene que antedecer al nombre de la instrucción la letra “E”, ejemplo: EMUL, EDIV.

7.5 Instrucciones de Comparacion

Este tipo de instrucciones se comportan como un contacto logico, que permite o no la ejecucion de una instrucción o grupo de instrucciones, evaluando su condicion planteada.

todas las comparaciones aritmeticas posibles estan disponibles en los PLC Cimon:



< S1 S2 si S1 es menor que S2
<= S1 S2 si S1 es menor o igual que S2
<> S1 S2 si S1 es diferente que S2
= S1 S2 si S1 es igual que S2
> S1 S2 si S1 es mayor que S2
>= S1 S2 si S1 es mayor o igual que S2

En este tipo de instrucciones solo podemos comparar variables enteras, dobles enteras y Float.

tambien podemos alternar el tipo de variable a operar, entero, doble entero o numero flotante.

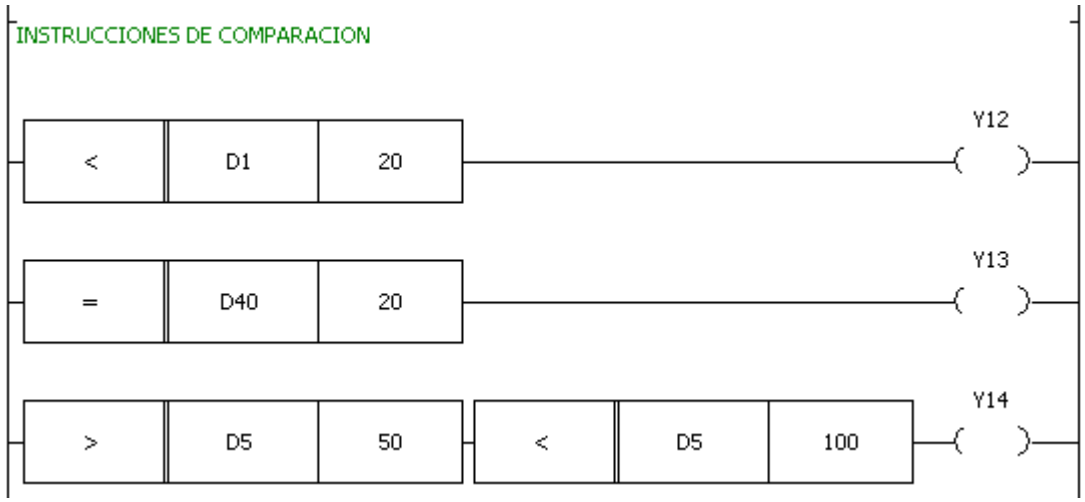
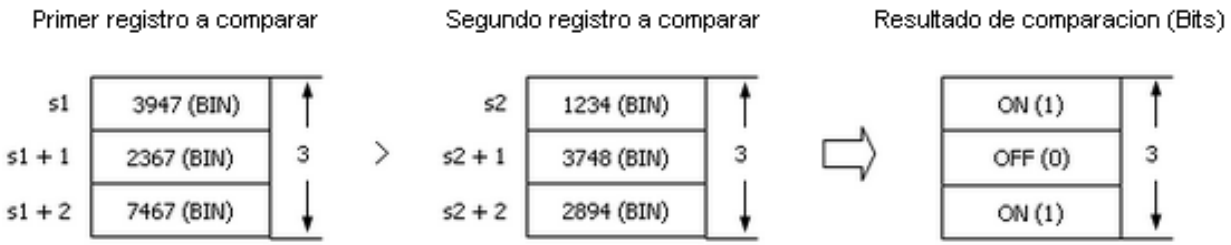


Figura 43: instrucciones de comparacion

Existe tambien una instrucción BK que hace posible comparar varios registros a la vez, este tipo de instrucción se comporta como una instrucción de salida, y necesita un conctacto de condicionamiento para su implementacion.

El formato de escritura para esta instrucción es la siguiente:

BKxx S1 S2 S3 offset #data
xx: es el signo de la comparacion a realizar (>,>=,<>,<,<=,=)
S1: primer registro a comparar.
S2: segundo registro a comparar.
S3: Registro de resultado de comparacion, (bits internos).
offset: offset de numero de bits en resultado de comparacion.
#data: numero de registros de comparacion



En el siguiente ejemplo se observa que los registros D100 al D105 seran comparados con los registros D110 al D115.
El resultado de su comparacion se indicara en los bits a partir de M100.. a M105, la funcion de comparacion es la de igualdad.

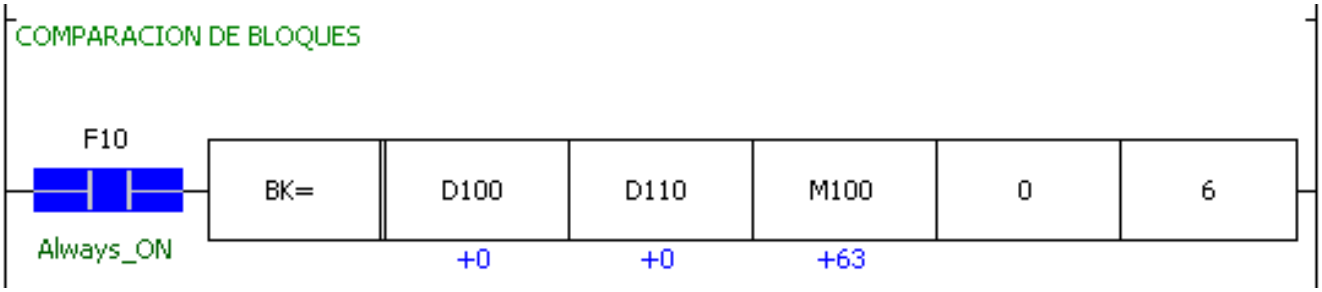


Figura 44: instrucción de comparacion de bloque

7.6 Instrucciones de Temporizadores y Contadores

los temporizadores y contadores en los PLC, son muy conocidos y usados. El PLC cimon posee las temporizaciones clasicas como TON, y TOF. Asi tambien los contadores de incremento y decremento.

7.6.1 Temporizadores

En el PLC cimon las variables del tipo T son reservadas para el manejo de las instrucciones con temporizadores.

Estas variables pueden representar el valor del acumulador de temporizacion o tambien el estado logico del fin de la cuenta en temporizacion según sea el uso de la variable en funcion de la instrucción.

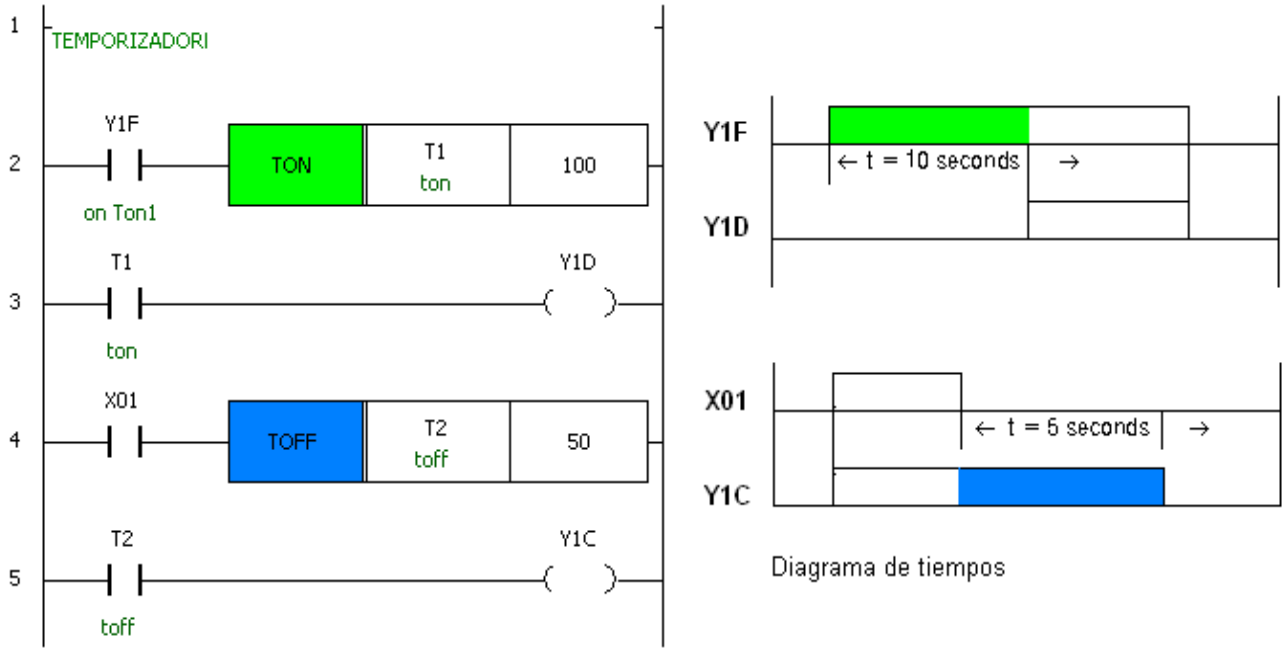


Figura 45: Instrucciones de temporizacion TON y TOFF

Las constantes e intervalo de tiempo de las temporizaciones estan definidas en el numero de registro T, utilizado, existen 2 tiempos de muestreo, cada 100ms y cada 10ms.

para poder ver que intervalo usar tenemos que verificar el menu de parametros configurados en el PLC, (ver figura siguiente), y seleccionar el rango de numeros de registros que presentaran el muestreo de 100ms.

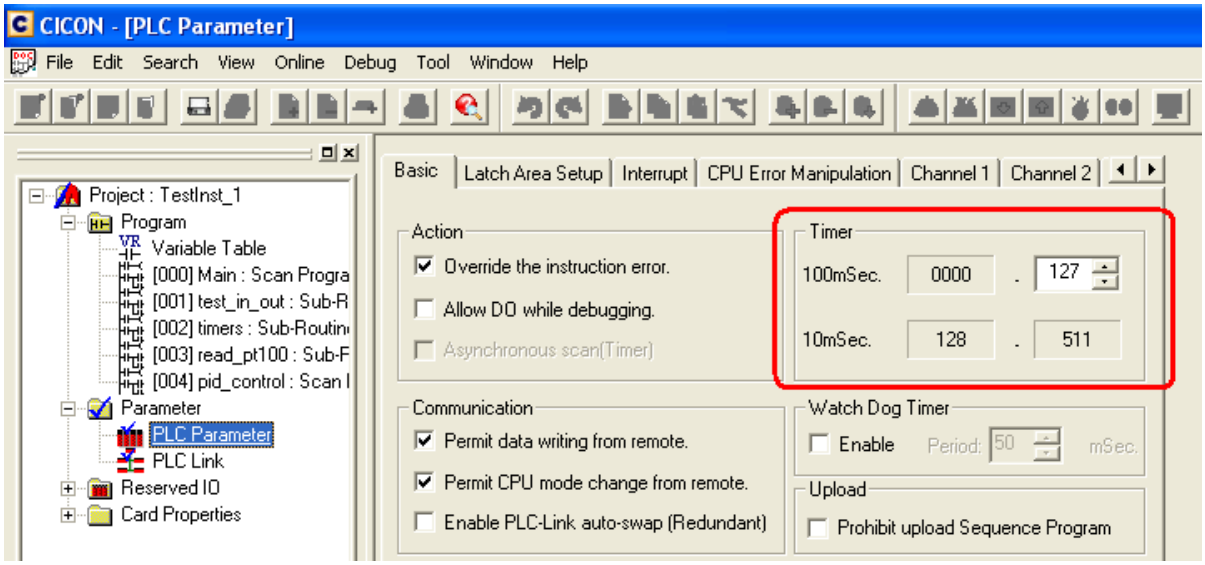


figura 46: Menu de parametros de PLC, indicando programacion de timers.

En el grafico anterior se denota que los registros T000 al T0127 tendran el muestreo de cada 100ms, y a partir del registro T128 al T511, el muestreo sera cada 10ms.

esto quiere decir que si se utiliza la instrucción Ton con un valor de preset de 50, con el registro T2, el tiempo programado de temporizacion (Toff) sera de 50*Tmuestro = 5000mS = 5 Segundos. Ejemplo: TON T2 50.

El tiempo de retardo es el valor acumulado x el tiempo de muestreo

Existen ademas de las instrucciones TON y TOFF otras instrucciones de temporizacion tales como TMR, TRTG, TMON, revisar el manual de ayuda de CICON para mayor informacion.

7.6.2 Contadores

Existen 3 contadores basicos CTU (incremental) CTD (decremental), ambos utilizan las variables tipo “C”, incluyen los contactos clasicos de reset.

A continuacion se muestra un diagrama de su funcionamiento:

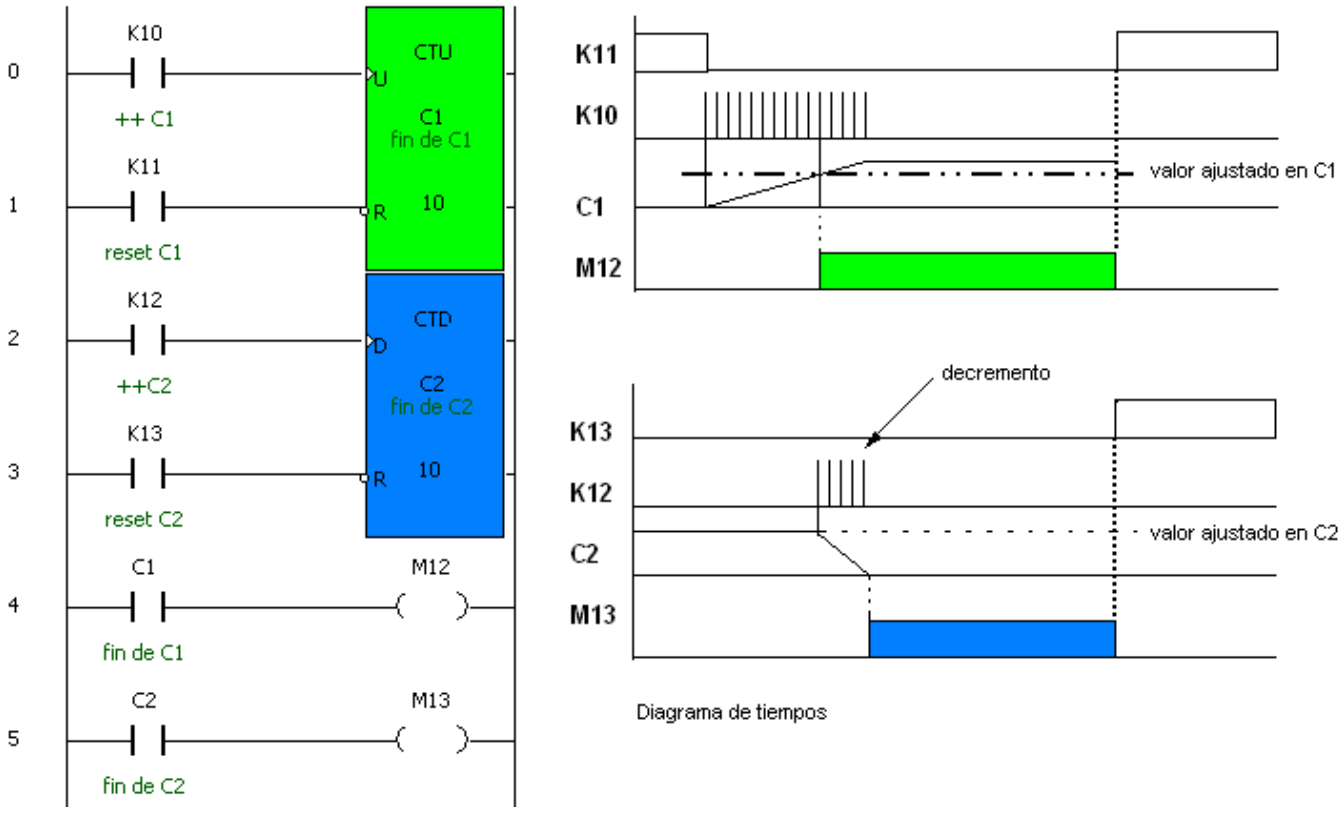


Figura 47: Instrucciones de Contador CTU y CTD

cuando se generen impulsos sobre el terminal U/D de los contadores CTU/CTD la cuenta registrada en la variable C1 o C2 se incrementara o decrementara respectivamente.

- Cuando en un contador CTU, el valor acumulado en Cx alcanza el valor de set point, el bit Cx permanece activo.
- Cuando en un contador CTD, el valor acumulado en Cx alcanza el valor 0, el bit Cx permanece activo.
- La entrada reset de contador (R) permite ajustar el valor de inicio por defecto del contador, 0 para CTU y set point para CTD.

Existen otros contadores tales como CTUD, que es la combinacion de un contador CTU y CTD, para mayor informacion revisar el manual de ayuda de CICON.

7.7 Instrucciones de operaciones logicas

Estas instrucciones permiten aplicar las funciones logicas conocidas tales como AND, OR, XOR y sus combinaciones posibles en aplicación de registros y bloques de registros.

Las opciones posibles son las siguientes:

Funcion	Instrucción		
Logica	word	doble word	bloque registros
and	WAND	DAND	BKAND
or	WOR	DOR	BKOR
xor	WXOR	DXOR	BKXOR
xor negado	WXNR	DXNR	BKXNR

La fila Word muestra las instrucciones logicas para variables del tipo word.
Ejemplo: WAND D5 D7 D10

Las instrucciones de operaciones logicas son del tipo instrucciones de salida, lo que quiere decir que no condicionan la ejecucion del reglon actual de la linea de programa ladder donde se encuentra.

la nomenclatura de las instrucciones es la siguiente:

[Funcion logica word/Dword] [S1] [S2] [Ds]

Donde S1 y S2 son los registros a comparacion bit a bit de la operación
y Ds es el destino y resultado de la operación logica realizada (bit a bit).

[Funcion logica bloque registro] [S1] [S2] [Ds] [n]

Donde S1 y S2 son los registros a comparacion bit a bit de la operación
Ds es el destino y resultado de la operación logica realizada (bit a bit).
n es el numero de registros a partir de S1 y S2 en la cual se realizara la comparacion.

En el grafico siguiente se podra observar la operación AND entre bloque de registros con mayor detenimiento:

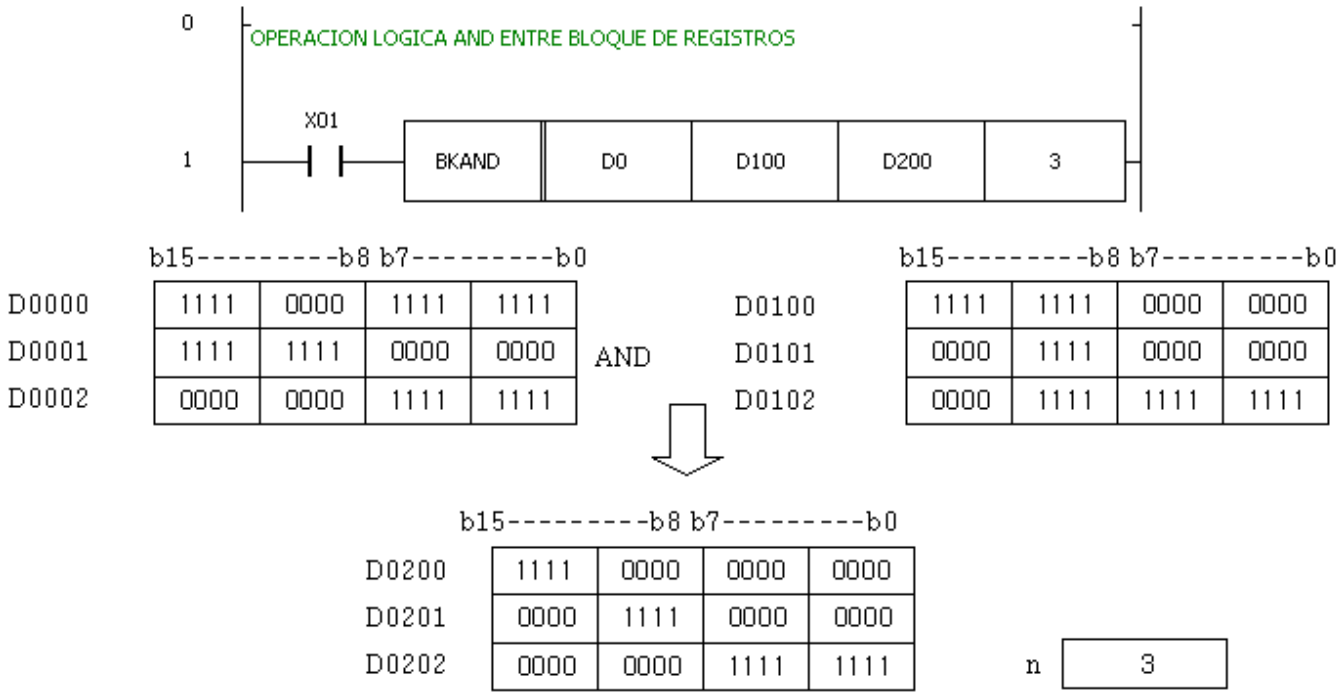


Figura 48: operación logica AND entre bloque de registros.

7.8 Llamadas de subrutina

las instrucciones de control de flujo de programa son muy importantes porque permiten distribuir la ejecucion de las tareas estableciendo un perfil o una rutina que permita la eficiencia de la ejecucion de las tareas asi como el orden de ejecucion tando en la aplicación como en el mismo desarrollo del proyecto.

La programacion extructurada dentro de los lenguajes de programacion existentes presentan gran ventaja respecto a los otros clasicos metodos de programacion en la que la extructura del programa obedece a un script principal sin requerir saltos de programa o llamadas a subprocesos dentro de programa principal.

Esto permite la distribucion eficiente del codigo asi comon el ahorro de uso de memoria en las tareas que se repetiran frecuentemente y se consideraran como tareas de “rutina”.

Los PLC Cimon, poseen 3 tipos de salto de programa:

Salto incondicional: JMP,JME
Llamada subrutina dentro de programa: CALL
Llamada subrutina entre diferentes programas: ECALL

El uso del salto incondicional no es recomendado, si no se tiene los conceptos claros de la distribucion de la memoria de programa y como evitar el cazo de bucles sin retorno.

Las instrucciones mas utilizadas y recomendadas son las llamadas a subrutina.

7.8.1 instrucción ECALL

esta instrucción permite llamar a una subrutina que se encuentre fuera del programa actual de ejecucion, este programa en ejecucion puede ser tambien una subrutina , que añadira en este caso un nivel mas de salto sobre la pila de memoria STACK, del PLC Cimon.

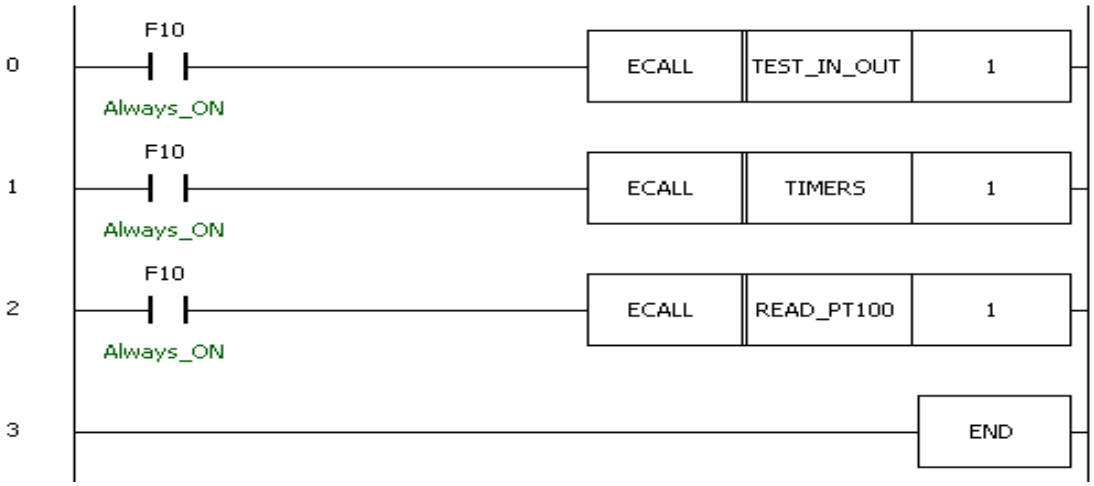


figura 49: fila de llamadas de subrutina ECALL

En el ejemplo anterior se muestra que se utiliza las llamadas de subrutina a los programas, test_in_out, timers, read_pt100, adicionalmente se coloca una constante 1, que indica que dentro de estas subrutinas se estara ejecutando el bloque de programa con etiqueta 1.

En la figura de la derecha se observa la extructura del programa test_in_out.
en este programa se etiqueta 2 areas de programa con la instrucción SBRT, esta instrucción separa los parte del programa hasta la ubicación de la instrucción RET, que cumple la funcion de retorno de llamado de subrutina.

de esta forma el reglon 6 de la derecha no se ejecutara si se utiliza el llamado de subrutina mostrado en la figura 49.

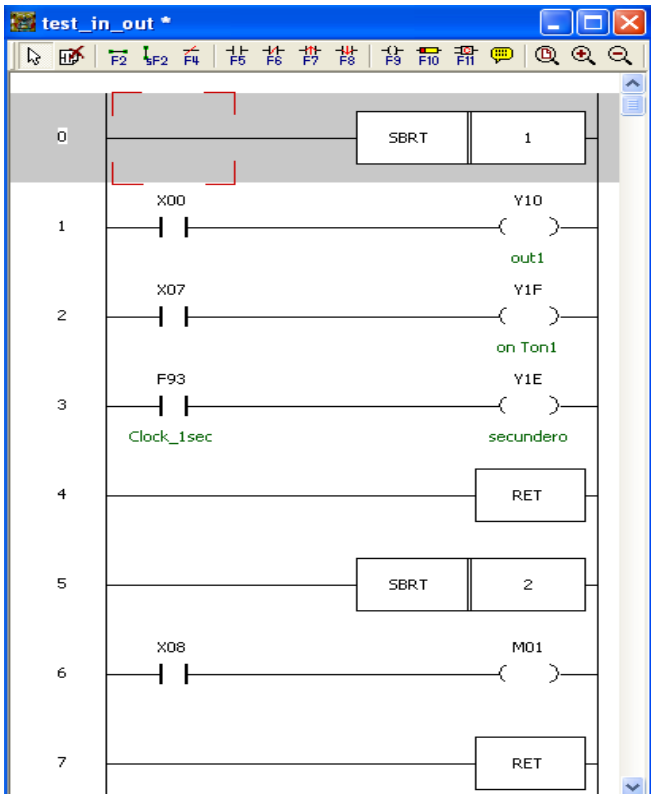


Figura50: subrutina de programa

- La instrucción SBRT se utiliza para etiquetar parte del programa que iniciara la ejecución de las instrucciones después de un llamado de subrutina que indexe la misma etiqueta. Esta instrucción se coloca siempre individualmente en una sola línea.
- La instrucción RET, se coloca para retornar al programa anterior o ubicación anterior de la memoria de programa donde se realizó el último llamado de subrutina. Esta instrucción no debe utilizarse en serie con otra instrucción.
- Cuando un código de subrutina se encuentre dentro de la misma hoja de programa solo es necesario usar la instrucción Call, que ejecutara dicha subrutina indexada con una etiqueta numérica.

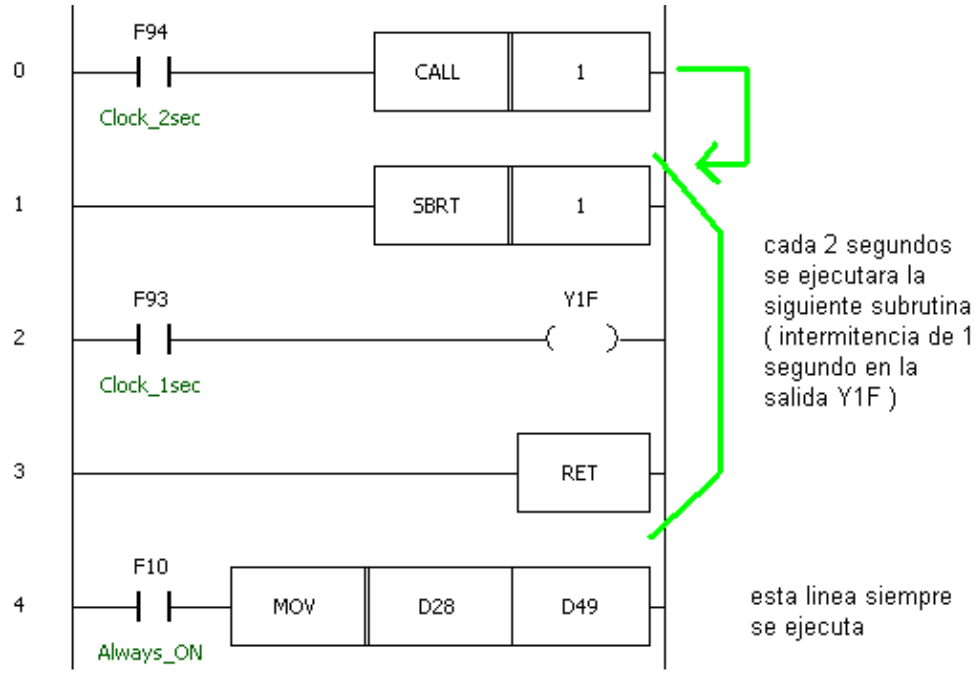


Figura 51: uso de la instrucción Call en hoja de programa.

7.8.2 Creacion de programa subrutina

Para crear un programa que se ejecute como subrutina debemos agregar un nuevo programa sobre el arbol de proyecto:

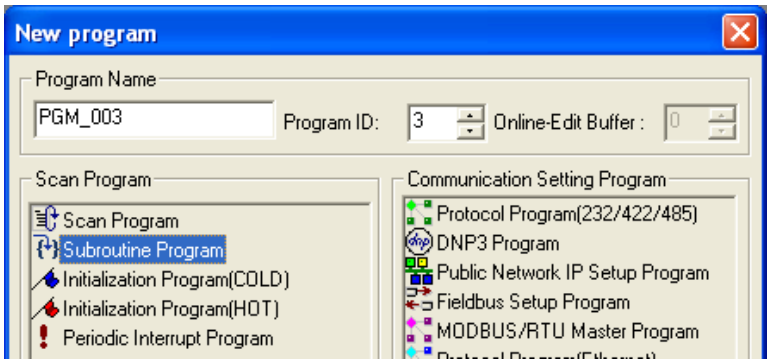


Figura 52: agregar un programa en modo subrutina.

Este programa ya no contara con la instruccion Final END, ya que es solo una subrutina de ejecucion.

- Es necesario primero crear el programa de subrutina antes de colocar una instrucción de llamado sobre el programa principal.
- Para hacer los llamados de subrutina se tiene que utilizar un solo reglon, por ende se debe evitar colocar dos instrucciones de llamado de subruitna en paralelo.

7.9 Instrucciones de salida con pulso

La mayoria de instrucciones de salida pueden ejecutarse agregando la funcion de deteccion de pulso, esto quiere decir que si la linea actual de programa se activa por una sola vez, originando un flanco de subida sobre la instrucción de salida, esta instrucción solo ejecutara en ese momento, no importando si la linea actual de programa permanece activa.

Para agregar esta opcion adicional a las instrucciones de salida se le agrega al nombre de la instrucción correspondiente la letra “P”,

Ejemplo: WANDP, MOVP , ECALLP, ADDP, BCDP, INTP, RCRP, MAXP, DATERDP, etc.

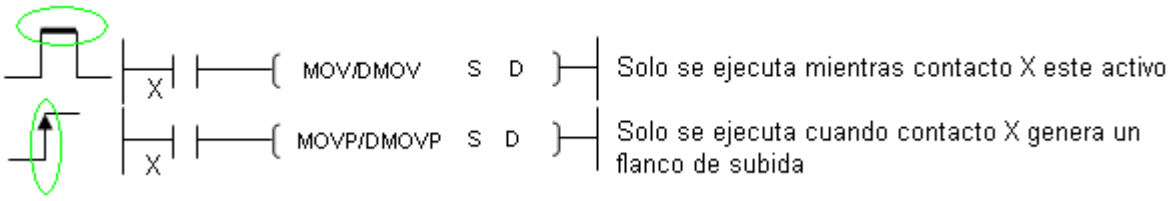

X
MOV/DMOV S D
Solo se ejecuta mientras contacto X este activo
X
MOVP/DMOVP S D
Solo se ejecuta cuando contacto X genera un flanco de subida

Figura 53: diferencias entre instrucciones mov y movP

Este forma de ejecutar la instrucción permite que solo se ejecute dicha instrucción en un solo instante de tiempo y no constantemente mientras un contacto de entrada este activo, esto puede tener ventajas cuando se utiliza instrucciones matemáticas, o rutinas que requieran gran cantidad de tiempo de ejecución, y no se desea que se ejecute constantemente en función de la activación de un contacto de entrada.

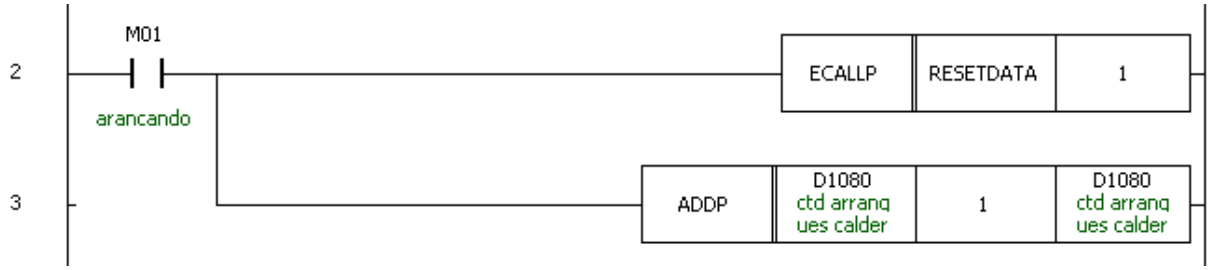


Figura 54: utilizacion de instrucciones de impulso

8. MANEJO DE MODULOS DE EXPANSION

Para el manejo de los módulos de expansión del PLC es necesario conocer la forma correcta de direccionar la memoria compartida tanto del PLC como de los módulos.

Dependiendo del modelo del PLC Cimon existen diferentes módulos, que a su vez difieren en la dirección de memoria a leer.

Para el caso de los PLC serie S, los siguientes módulos están disponibles:

PLCS AD Module : SP04EAO	módulo análogo de entrada con 4 canales Voltaje o corriente
PLCS DA Module : SP04EOAV	módulo análogo de salida con 4 canales (voltaje)
PLCS DA Module : SP04EOAI	módulo análogo de salida con 4 canales (corriente)
PLCS AD DA Module : SP04EAA	módulo análogo mixto de entrada y salida con 4 canales de voltaje o corriente
PLCS RTD Module : SP04ERO	módulo análogo para conexión de RTD, 4 canales
PLCS TC Module : SP04ETO	módulo análogo para conexión de termocupla, 4 canales
PLCS AD MUX Module : SP04EAM	módulo análogo para multiplexion de entradas análogas
PLCS SP32EDO	modulo digital de 32 entradas
PLCS SP32EOT	modulo digital de salida, 32 tipo transistor
PLCS SP32EDT	modulo digital mixto, 16 entradas / 16 salidas
PLCS SP16EOR	modulo digital, 16 salidas tipo Relay
PLCS SP01EET	módulo de comunicación Ethernet
PLCS SP02ERS	módulo de comunicación serial RS232/RS422/RS485
PLCS SP02ERR	módulo de comunicación serial RS232, 2 canales

8.1 Lectura de modulos Digitales de expansion.

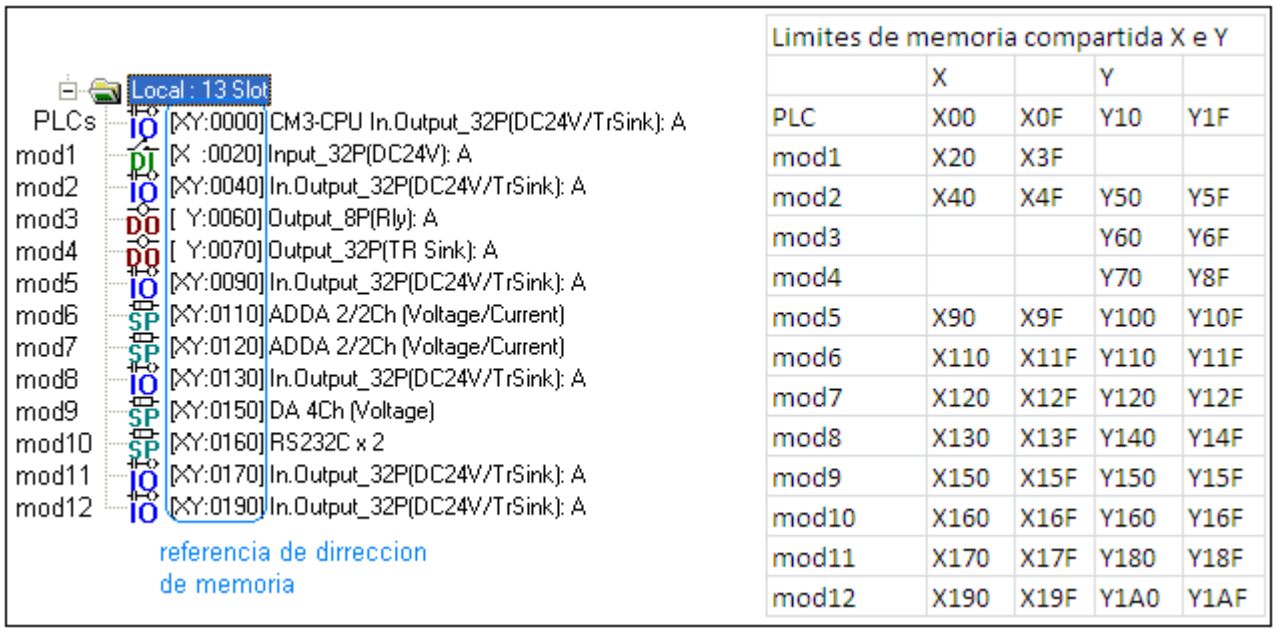
Cuando es necesario utilizar mas datos digitales desde el PLC, es necesario acoplar un modulo de expansion Digital. El numero total de modulos conectados al plc son hasta 11.

Para poder entender claramente el manejo de estos modulos, (Lectura / Escritura), es necesario entender de que manera la memoria de datos del PLC se divide para mostrar las señales de entrada y salida digitales de los modulos conectados a el.

La memoria asociada del PLC, son del tipo X e Y. esta memoria es compartida por todos los modulos de expansion, por lo tanto es necesario conocer cual es forma correcta de dirreccionamiento.

En el siguiente grafico se observa la distribucion de los modulos de expansion sobre un PLC de la serie S, Ahora para poder saber a que dirreccion apuntar sobre las entradas 'X' e 'Y' de los modulos de expansion es necesario fijarse en el recuadro celeste que se indica como "referencia de dirreccion de memoria"

Esta dirreccion representa el indice inicial desde donde se debe dirreccionar los datos digitales.


Local : 13 Slot
PLCs [XY:0000] CM3-CPU In.Output_32P(DC24V/TrSink): A
mod1 [X :0020] Input_32P(DC24V): A
mod2 [XY:0040] In.Output_32P(DC24V/TrSink): A
mod3 [Y:0060] Output_8P(Rly): A
mod4 [Y:0070] Output_32P(TR Sink): A
mod5 [XY:0090] In.Output_32P(DC24V/TrSink): A
mod6 [XY:0110] ADDA 2/2Ch (Voltage/Current)
mod7 [XY:0120] ADDA 2/2Ch (Voltage/Current)
mod8 [XY:0130] In.Output_32P(DC24V/TrSink): A
mod9 [XY:0150] DA 4Ch (Voltage)
mod10 [XY:0160] RS232C x 2
mod11 [XY:0170] In.Output_32P(DC24V/TrSink): A
mod12 [XY:0190] In.Output_32P(DC24V/TrSink): A
referencia de dirreccion de memoria

Limites de memoria compartida X e Y

	X		Y	
PLC	X00	X0F	Y10	Y1F
mod1	X20	X3F		
mod2	X40	X4F	Y50	Y5F
mod3			Y60	Y6F
mod4			Y70	Y8F
mod5	X90	X9F	Y100	Y10F
mod6	X110	X11F	Y110	Y11F
mod7	X120	X12F	Y120	Y12F
mod8	X130	X13F	Y140	Y14F
mod9	X150	X15F	Y150	Y15F
mod10	X160	X16F	Y160	Y16F
mod11	X170	X17F	Y180	Y18F
mod12	X190	X19F	Y1A0	Y1AF

Figura 55: Obtencion de Distribucion de memoria de variables X e Y para modulos Cimon

En la table de limites de memoria compartida X e Y, se observa que los contactos de salida siempre comienzan desde la dirreccion indice inicial, mas 1 palabra (+1Word), exeptuando los modulos especiales y los modulos de salida relay.

Para poder entender mejor la distribucion de memoria vamos a plantear un ejemplo, de configuracion de hardware:

	Modulo digital	entradas	salidas
PLC	PLCS CM3-SP32	16	16
Modulo 1	PLCS SP32EDO	32	
Modulo 2	PLCS SP32EDT	16	16
Modulo 3	PLCS SP16EOR		16
Modulo 4	PLCS SP32EOT		32
Modulo 5	PLCS SP32EDT	16	16

Esta aplicación constara de 1 PLC serie S que consta de 16 I/O digitales, al cual para poder expandir sus entradas y salidas se han incorporado 5 modulos I/O.

Ahora la disposicion de memoria para estos modulos sera la siguiente:

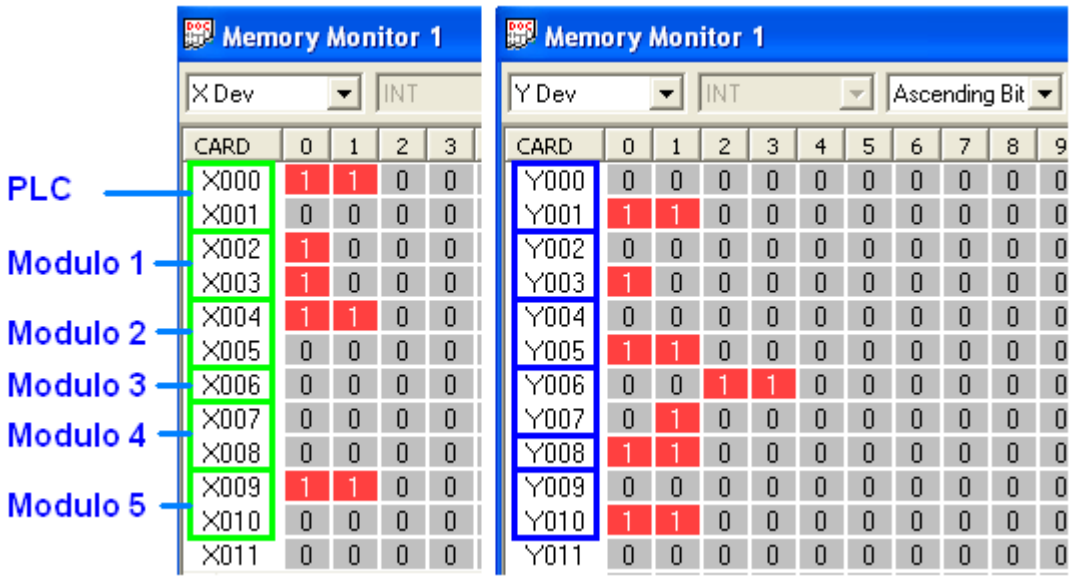


Figura 56: Distribucion de memoria de variables X e Y para modulos Cimon

Como se puede observar, por cada modulo se ocupan 2 words de memoria exepto el modulo especial de salida Rele que consta solo de 16 salidas digitales.

Otro detalle a tener en cuenta es que las salidas digitales fisicas del PLC o de los modulos Digitales de expansion, solo cuando son 16 se dirreccionan desde la seguna 'Word',

Por ejemplo las salidas Digitales del PLCS CM3-SP32MDTF Están implementadas sobre el CPU desde el Y10 al Y1F

Cada bloque de 32 salidas I/O pertenece a un diferente modulo.

Es decir :

Slot0 :	**y00 – Y1F**	**X00 – X1F**
Slot1:	**y20 – Y3F**	**X20 – X3F**
Slot2:	**y40 – y5f**	**X40 – X5F**

Si un bloque presenta 16 salidas I/O, solo este slot ocupa una palabra de memoria (1 word) ejemplo: el modulo PLCS SP16EOR

Para poder leer y escribir sobre estas entradas y salidas dentro del programa del PLC solo basta dirreccionar su dirreccion de memoria dentro del programa:

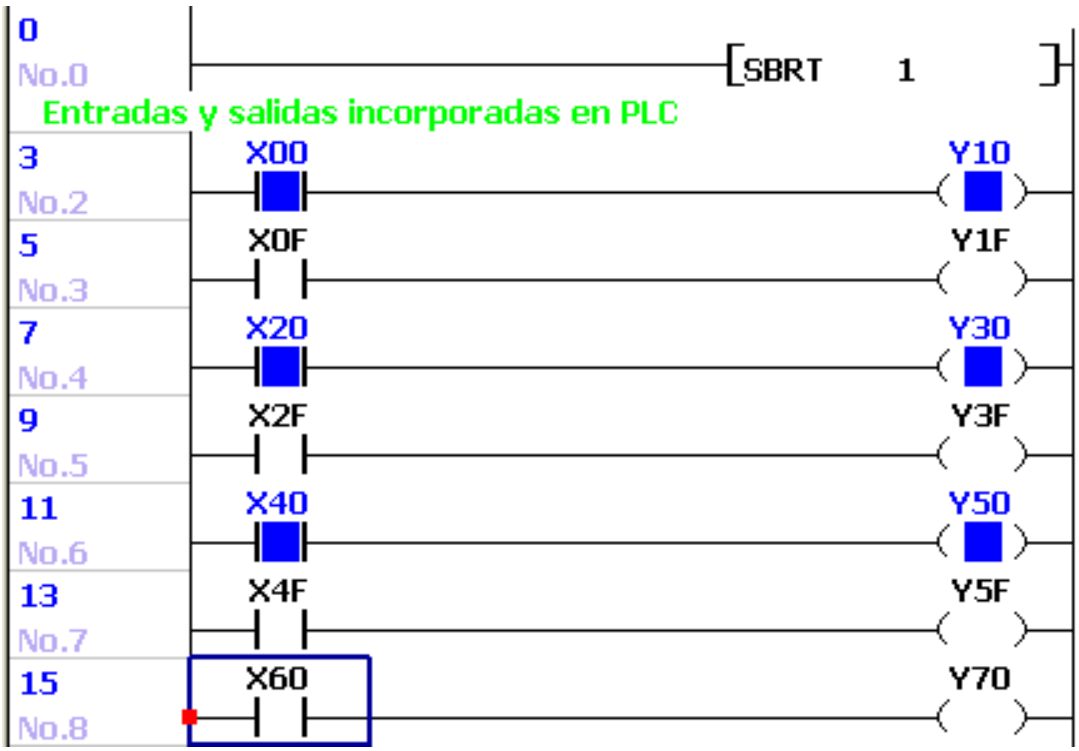


Figura 57: Programa Ladder en Editor nuevo de Cimon.

8.2 Lectura de modulos analogos.

En los PLC Cimon las entradas analogas se leen a traves de una instrucción, que toma datos directamente de la memoria interna del mismo modulo.

Estos modulos analogos tambien presentan bits internos de control que pueden dirreccionarse de la misma forma que los modulos I/O Digitales,

La distribucion de memoria para sus bits internos, esta separada en bloque de 1 palabra,

Para una mejor demostracion vizualizar el siguiente diagrama:

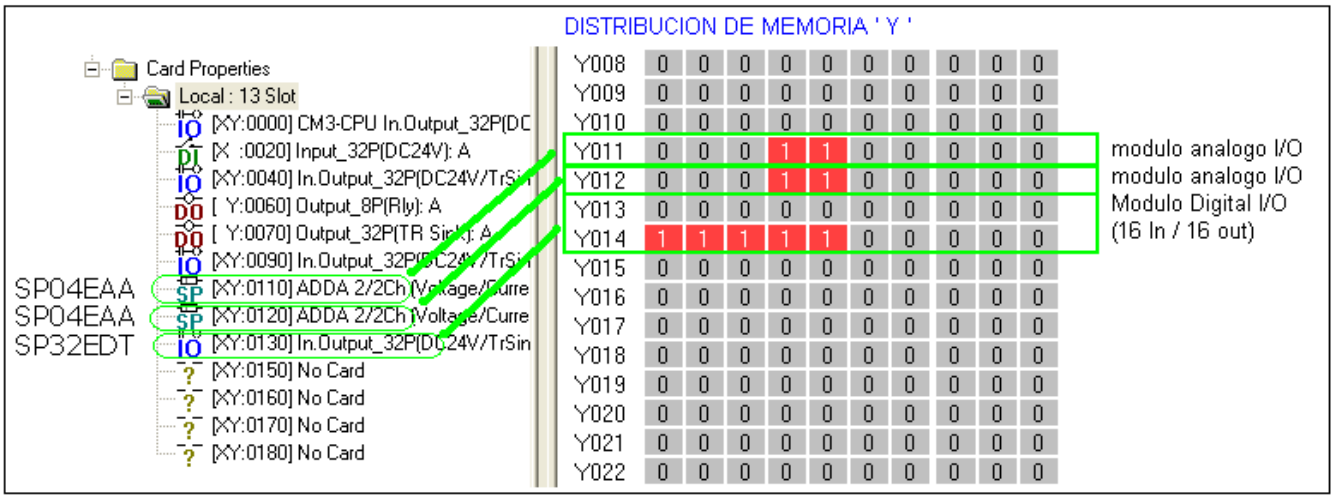

DISTRIBUCION DE MEMORIA ' Y '
Card Properties
Local : 13 Slot
[XY:0000] CM3-CPU In.Output_32P(DC
[X :0020] Input_32P(DC24V): A
[XY:0040] In.Output_32P(DC24V/TrSi
[Y:0060] Output_8P(Rly): A
[Y:0070] Output_32P(TR Sink): A
[XY:0090] In.Output_32P(DC24V/TrSi
SP04EAA [XY:0110] ADDA 2/2Ch Voltage/Curre
SP04EAA [XY:0120] ADDA 2/2Ch Voltage/Curre
SP32EDT [XY:0130] In.Output_32P(DC24V/TrSin
[XY:0150] No Card
[XY:0160] No Card
[XY:0170] No Card
[XY:0180] No Card
Y008 0 0 0 0 0 0 0 0 0 0
Y009 0 0 0 0 0 0 0 0 0 0
Y010 0 0 0 0 0 0 0 0 0 0
Y011 0 0 0 1 1 0 0 0 0 0
Y012 0 0 0 1 1 0 0 0 0 0
Y013 0 0 0 0 0 0 0 0 0 0
Y014 1 1 1 1 1 0 0 0 0 0
Y015 0 0 0 0 0 0 0 0 0 0
Y016 0 0 0 0 0 0 0 0 0 0
Y017 0 0 0 0 0 0 0 0 0 0
Y018 0 0 0 0 0 0 0 0 0 0
Y019 0 0 0 0 0 0 0 0 0 0
Y020 0 0 0 0 0 0 0 0 0 0
Y021 0 0 0 0 0 0 0 0 0 0
Y022 0 0 0 0 0 0 0 0 0 0
modulo analogo I/O
modulo analogo I/O
Modulo Digital I/O
(16 In / 16 out)

Figura 58: Mapa de distribucion de memoria para los modulos analogos.

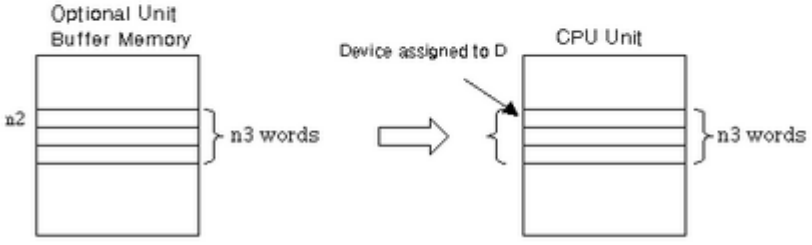
De igual manera las entradas X se podran dirreccionar con la misma dirreccion indice mostrada en el arbol de modulos mostrados.

8.2.1 Instrucción de lectura de datos de Modulos de expansion.

La instrucción para leer un dato word, desde un modulo de expansion, tal es el caso de los modulos analogos, se Llama FROM.

FROM

Esta instruccion es usada para leer 'n3' datos en la dirreccion asignada en 'n2', desde un buffer de memoria (Modulo de expansion) hacia la dirreccion asignada 'D' sobre la memoria del PLC



La nomenclatura de la funcion FROM es la siguiente.

FROM N1 N2 D N3

Donde:
N1 es el número del slot del módulo asociado de lectura.
N2 es la dirección de memoria del módulo de lectura.
D es la dirección de registro del PLC donde se almacenara dicho valor
N3 es el número de registros enteros que se leerán del módulo asociado

En el siguiente ejemplo se muestra la utilizacion de la funcion From

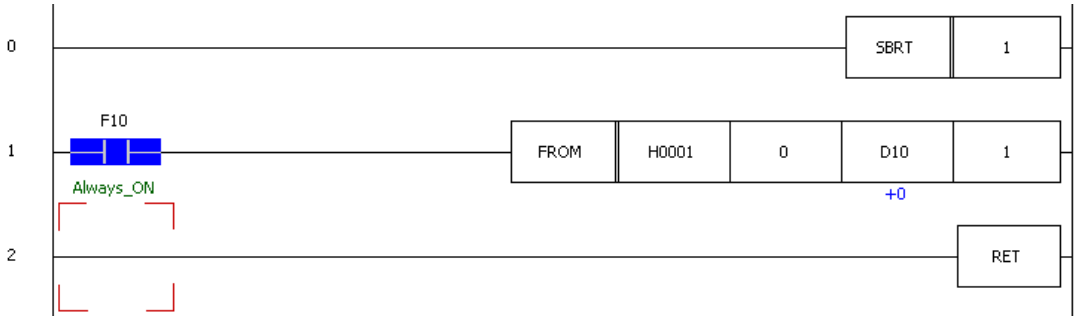
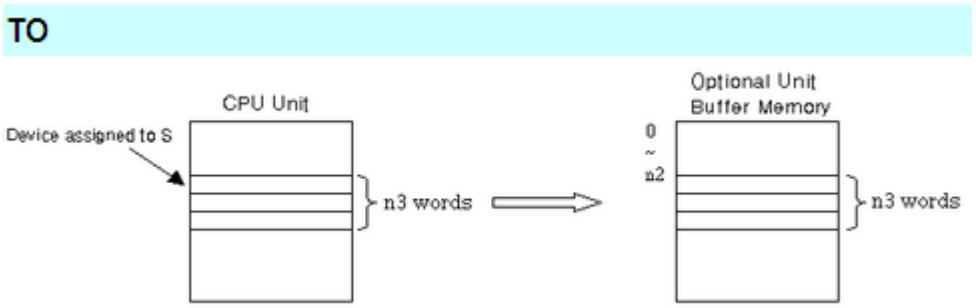


Figura 59: Uso de la instrucción From

En el anterior ejemplo se lee el registro 0x00 del slot #1 hacia la dirreccion D10 del PLC.

8.2.2 Instrucción de escritura de datos sobre Modulos de expansion.

Esta instrucción hace el procedimiento inverso de la funcion FROM



El formato de parametrizacion es similar, a la función FROM, la única diferencia radica en que el registro de memoria asociado al PLC, viene a ser el registro que contendrá el valor a escribir sobre el registro del módulo direccionado.

TO N1 N2 S N3

Donde:
N1 es el número del slot del módulo asociado de escritura.
N2 es la dirección de memoria del módulo de escritura.
S es la dirección de registro del PLC desde donde se copiara dicho valor
N3 es el número de registros enteros que se escribirán sobre el módulo asociado

8.2.3 Lectura de entrada analoga sobre modulo.

como se vio con anterioridad se tiene que utilizar las instrucciones FROM, para poder leer los datos almacenados sobre los modulos de expansion analogos.

Ahora es necesario saber ademas que dirreccion poder leer, esto se encuentra en la tabla de memoria de cada modulo que viene indicado en su manual u hoja de datos.

Para citar un ejemplo tomaremos el modulo SP04EAA

Address		Details	Default	R/W
Hex.	Dec.			
0H	0	CH.1 Digital conversion value (AD)	-	R
1H	1	CH.2 Digital conversion value (AD)	-	R
2H	2	CH.1 Precise value (AD)	-	R
3H	3	CH.2 Precise value (AD)	-	R
4H	4	CH.1 Percentile value (AD)	-	R
5H	5	CH.2 Percentile value (AD)	-	R
6H	6	High alarm status (AD)	-	R
7H	7	Low alarm status (AD)	-	R
8H	8	CH.1 Setup input signal range (AD)	0	R/W
9H	9	CH.2 Setup input signal range (AD)	0	R/W
AH	10	Setup Raw Value digital output (AD)	0	R/W
BH	11	CH.1 Average processing setting value (AD)	2000h	R/W
CH	12	CH.2 Average processing setting value (AD)	2000h	R/W
DH	13	CH.1 Maximum alarm setting value (AD)	0	R/W
EH	14	CH.2 Maximum alarm setting value (AD)	0	R/W
FH	15	CH.1 Minimum alarm setting value (AD)	0	R/W
10H	16	CH.2 Minimum alarm setting value (AD)	0	R/W
11H	17	CH.1 Digital Filter Constant (AD)	30	R/W
12H	18	CH.2 Digital Filter Constant (AD)	30	R/W
13H	19	DA Enable/Disable setup (DA conversion)	0	R/W
14H	20	CH.1 DA output type (Current 2types, Voltage 4types)	0	R/W
15H	21	CH.2 DA output type (Current 2types, Voltage 4types)	0	R/W
16H	22	CH.1 DA Range of digital value	0	R/W
17H	23	CH.2 DA Range of digital value	0	R/W
18H	24	CH.1 DA Channel Hold/Clear	0	R/W
19H	25	CH.2 DA Channel Hold/Clear	0	R/W
1AH	26	CH.1 digital output value (DA)	0	R/W
1BH	27	CH.2 digital output value (DA)	0	R/W
1CH	28	AD / DA Resolution setting	0	R/W
1DH	29	Error Code	-	R
1EH	30	OS Version	-	R

Figura 60: tabla de memoria del modulo analogo SP04EAA

Esta tabla tambien es llamada Buffer memory, sin embargo tambien existe otra tabla en la que se enumera la lista de entradas y salidas de control sobre el modulo analogo.

Direction of Signal(CPU←A/D D/A Module)		Direction of Signal(CPU→A/D D/A Module)	
Input	**Name of Signal**	**Output**	**Name of Signal**
X00	A/D Module Ready	Y00	
X01		Y01	
X02	Flag indicating the operation condition set	Y02	Requesting to set up an operation condition
X03	Alarma minimo en CH.1	Y03	Output Enable CH1(Control Command)
X04	Alarma maximo en CH.1	Y04	Output Enable CH2(Control Command)
X05	Alarma minimo en CH.2	Y05	
X06	Alarma maximo en CH.2	Y06	
X07		Y07	
X08		Y08	
X09		Y09	
X0A		Y0A	
X0B		Y0B	
X0C		Y0C	
X0D		YOD	
X0E		YOE	
X0F	Flag indicating an error in module	Y0F	Requesting to clear the error

Figura 61: Lista de entradas y salidas del modulo SP04EAA reflejadas sobre el PLC

En la figura 60, se observa que la dirreccion de lectura de entrada analogas puede ser dirreccionada desde 6 dirrecciones de memoria,

Valor de conversion Digital 0x00, 0x01
Valor de precision en unidades 0x02, 0x03
Valor de porcentaje. 0x04, 0x05

Apuntando a cualquiera de estos registros podremos obtener un lectura del canal analogo elegido, Ejemplo:

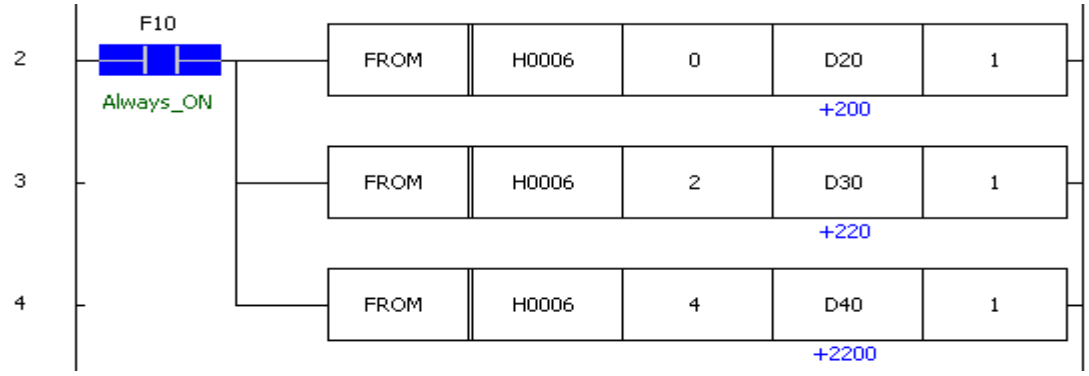


Figura 62:Instrucciones FROM para lectura de medidas analogas

En el anterior ejemplo se lee la señal analoga sobre el canal 1 del modulo del slot 6, todo esto mediante el modulo SP04EAA.

- El slot numero 0 le corresponde al PLC
- El slor numero 1 viene hacer el primer modulo de expansion conectado al PLC (de izquierda a derecha).

8.2.4 Escritura sobre salida analoga de modulo.

Para poder escribir sobre una salida analoga se utiliza la instrucción TO, similarmente que la instrucción FROM, se tiene que tener en cuenta que existen bits de control y habilitacion que permiten la salida analoga sobre el modulo aparezca.

estos bits de control pueden aparecer sobre la "buffer memory", o sobre la memoria compartida X e Y, del modulo sobre el PLC.

para el caso del modulo SP04EAA, los bits que habilitan la salida analoga son los bits Y04 y Y05 mostrados sobre la figura 60.

Ahora esta dirreccion tiene que ser sumada al indice de dirreccion del modulo analogo actual.

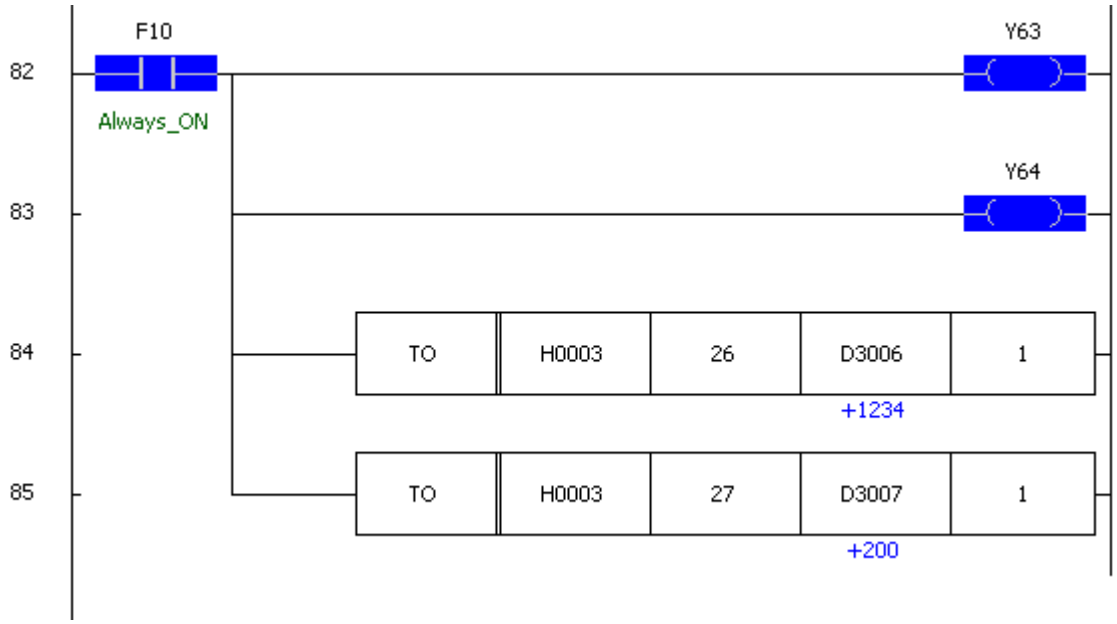


Figura 63: Instrucciones TO para escrituras de señales analogas.

En el ejemplo anterior, se desea escribir sobre las salidas analogas CH1 y CH2 del modulo SP04EAA, ubicado en el slot 3,
En el slot 1 y 2 se encuentra el modulo SP32EDT, que ocupa las salidas digitales y20-y3f ... y40-y5f.

Los contactos Y63 y Y64 del anterior ejemplo son los controles de habilitacion de salida analoga sobre los canales de salida del modulo. Es por eso que aparecen activos estos contactos dentro del programa.

En conclusion utilizando las instrucciones FROM/TO podemos leer todos los datos almacenados en los modulos de expansion, en especial los modulos Analogos.
solo es necesario tener presente la tabla o memoria de datos que clasifica los datos requeridos a solicitud.

9. COMUNICACIÓN CON DISPOSITIVOS EXTERNOS

El PLC Cimon presenta varias formas de comunicarse utilizando protocoles estandares y comunes de la mayoria de dispositivos.

Los puertos de comunicación estandares utilizados son ETHERNET, Serial RS232, Serial RS485.

Ademas existe un protocolo Propio de Cimon que permite la comunicación mas directa y eficaz con los equipos de la misma marca, HMI, PC con SCADA CIMON.

9.1 Comunicación con el HMI Cimon.

Cuando se establece la comunicación con un HMI, se puede utilizar el protocolo propietario de CIMON, esta configuracion principalmente se realiza sobre el HMI.

Sobre el PLC, solo se configura los parametros de RED cuando se utiliza la comunicación via Ethernet, y los parametros de comunicación serial, Baudios, paridad, bit de datos, cuando se utiliza la comunicación serie.

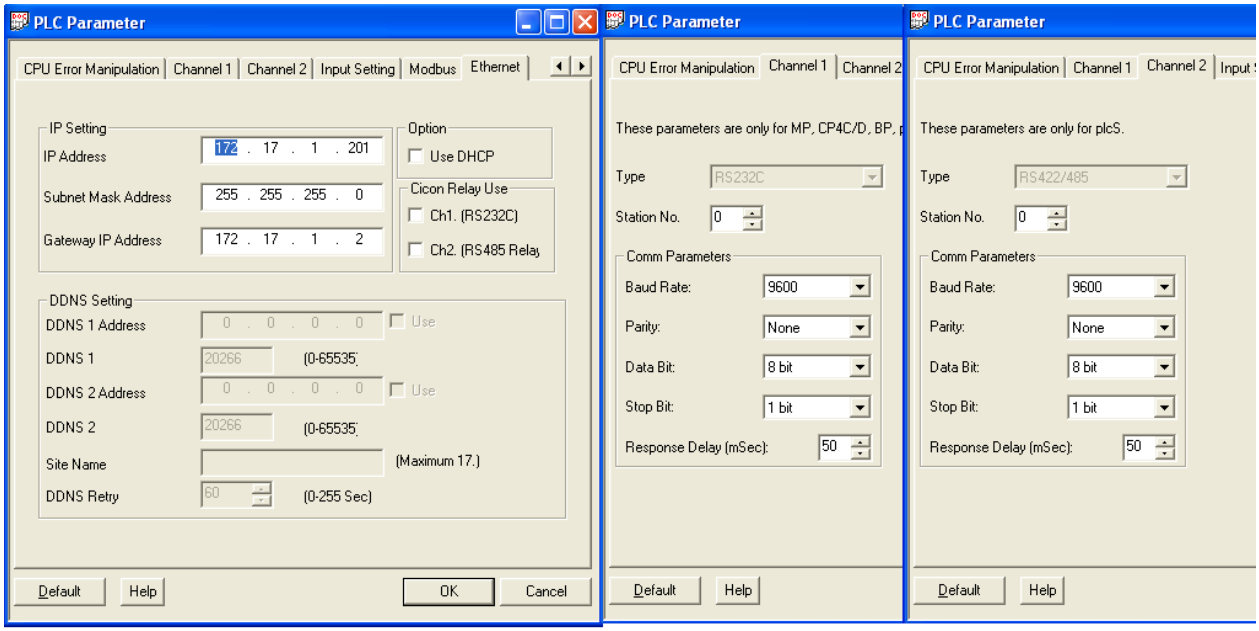


Figura 64: Ventana de parametros del PLC para comunicación Ethernet, serial RS232/RS485.

En la comunicación serie es necesario siempre indicar el numero de estacion al cual se ha configurado el PLC, esto sobre el menu de PLC parameter.

La forma de dirreccionamiento sobre el HMI, es igual a la nomenclatura de dirrecciones en el PLC.

El nombre del protocolo propietario de CIMON es KDT Systems CIMON-PLC LOADER.

Un detalle importante es que cuando se va a conectar un PLCs Cimon con un HMI Cimon sin utilizar algun SWITCH Ethernet, es necesario utilizar un cable Ethernet Cruzado.

Cuando se utiliza modulos de comunicación de expansion, es necesario tambien indicar el protocolo a utilizar:

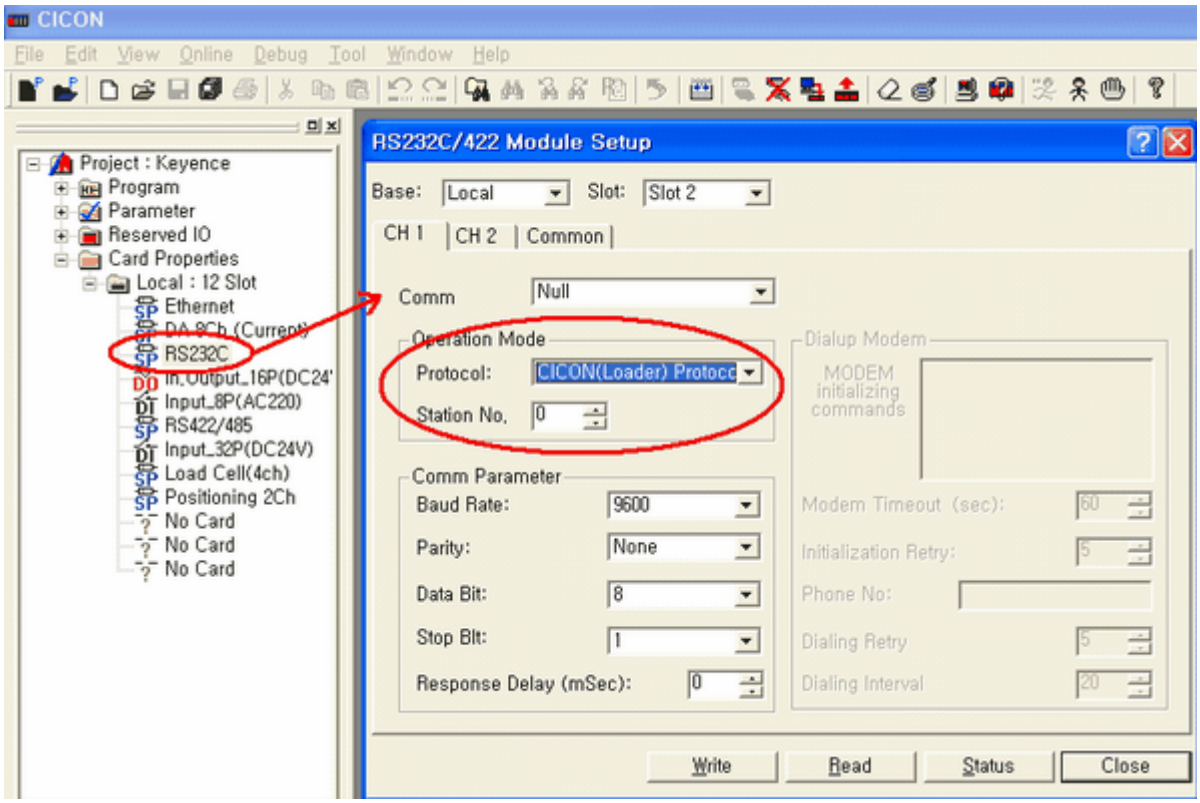


Figura 65: Ventana de parametros del PLC para comunicación Ethernet, serial RS232/RS485.

Para ver detalles de las conexiones electricas a realizar revisar el tema 11.

9.2 Comunicación utilizando el protocolo MODBUS

El protocolo modbus, es un protocolo generico ya antinguo pero muy usado aun en la actualidad por equipos remotos/esclavos de los cuales se desea recoger sus datos.

El PLC cimon posee los algoritmos de comunicación incorporados para comunicarse via Ethernet (Modbus TCP) y Serial (Modbus Rtu), utilizando las opciones de Maestro y Esclavo para ambas interfaces.

Cuando el PLC trabajara en modo MODBUS-Esclavo, solo es necesario configurar los parametros de comunicación correspondiente a la interfaz elegida, para finalmente, dirreccionar los datos según como indique el mapa de registro Modbus del PLC.

Para poder como dirreccionar los datos del PLC via modbus, nos dirigimos a la ventana de parametros del PLC, y sobre la pestaña Modbus, observaremos los indices de registro para leer/Escribir.

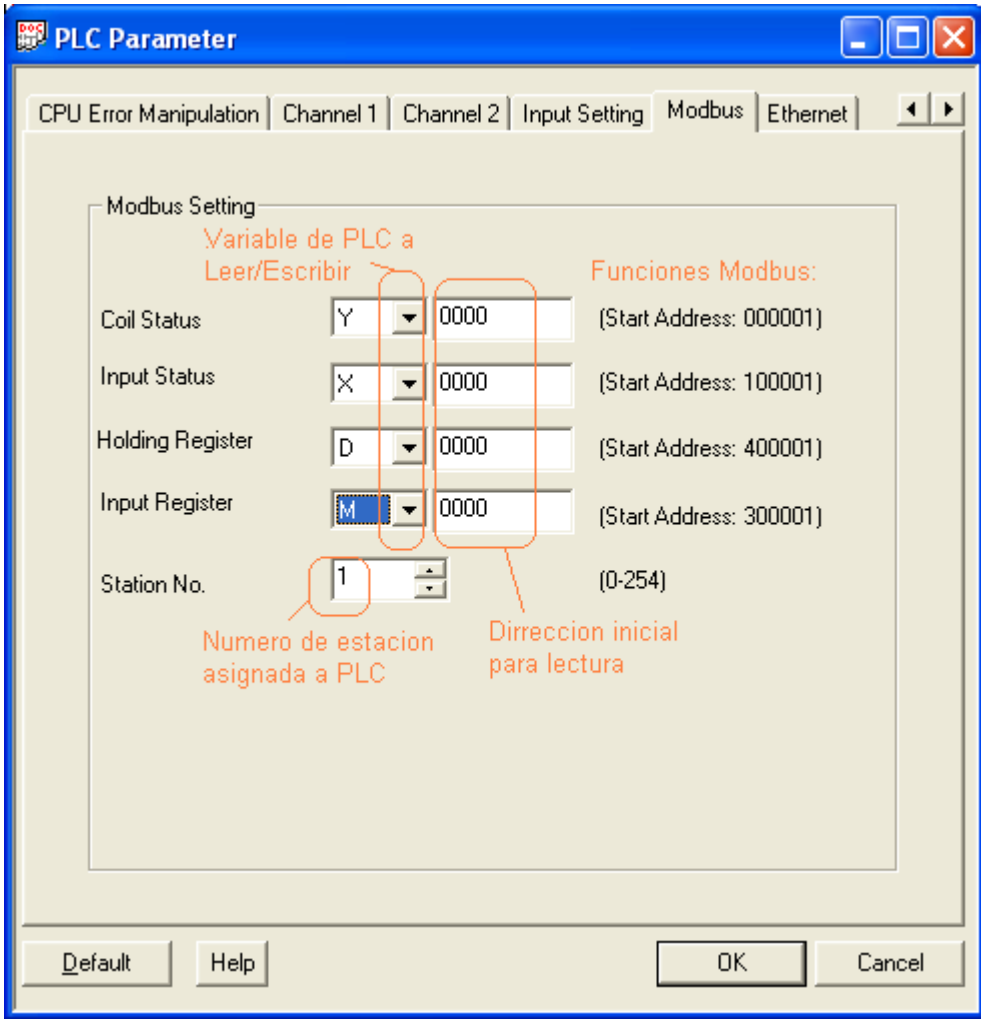


Figura 66: Mapa de registros Modbus.

En el anterior grafico se observa que para poder leer los registros tipo M, tenemos que utilizar la funcion 03, del protocolo modbus.

De igual manera para leer los registros tipo D, tenemos que utilizar la funcion 04.

la dirrecion de lectura viene en funcion a la relacion entre la columna que dice Dirreccion inicial para lectura, y el valor mostrado en Start Address.

Es decir según el grafico anterior, para leer el registro D0007, tenemos que usar la funcion modbus 04, dirreccionando el registro 0x08.

9.2.1 Comunicación via MODBUS RTU modo Maestro.

El software Cicon presenta una plantilla de programacion para captura de datos via Modbus, de esta forma la programacion utilizando este medio, se hace mas practica y flexible para las aplicaciones que se desee.

Para crear una plantilla de programacion para comunicación via Modbus, debemos crear un nuevo programa bajo las siguientes caracterisiticas mostradas en la siguiente figura:

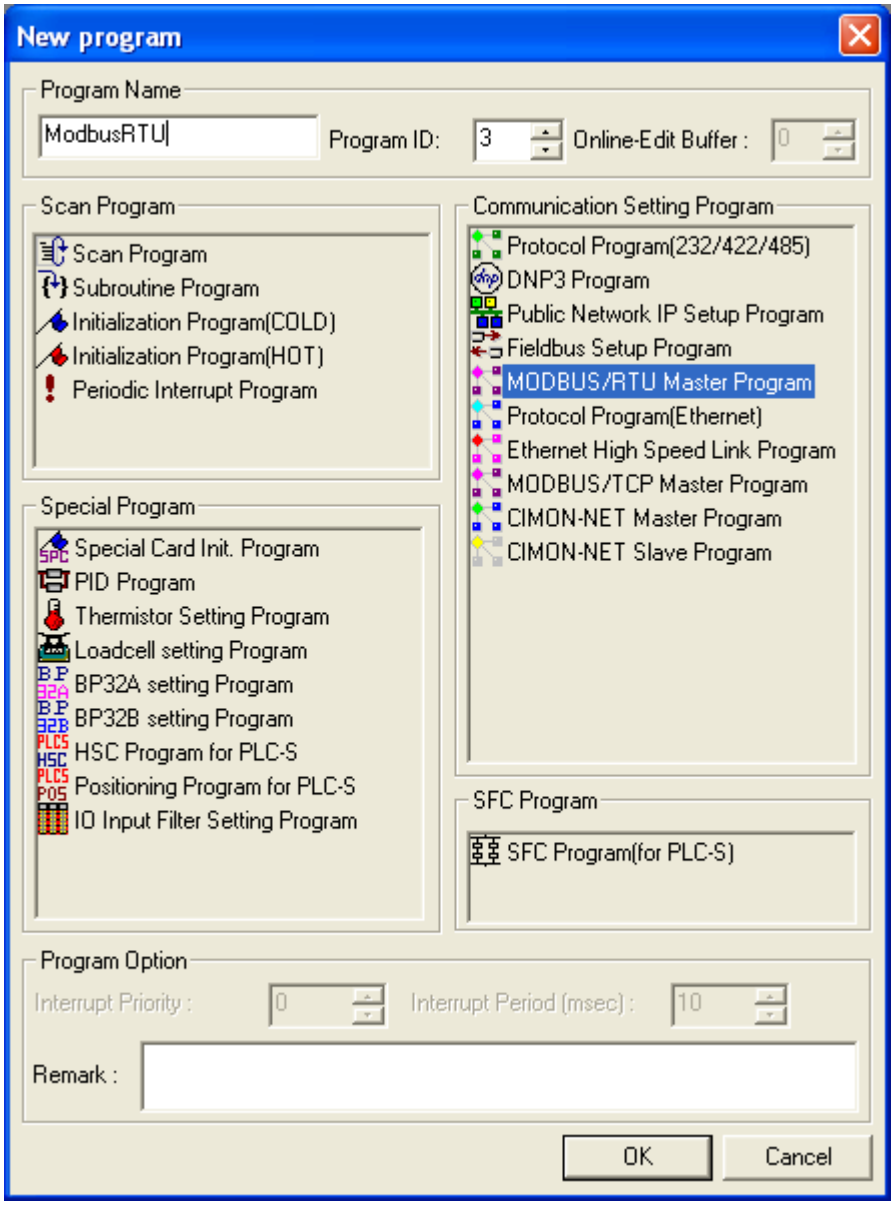


Figura 67: Creacion de programa plantilla Modbus RTU - Maestro.

En el grafico se ve que se tiene que seleccionar la opcion Marcada de azul (MODBUS/RTU Master Program).

Al crear este programa, se abrira una nueva ventana, donde transcribiremos los datos que deseemos obtener de los equipos interconectados.

Para agregar un nuevo link de enlace de dato a leer o escribir, tenemos que presionar el boton Insert, luego aparecera una pequeña ventana desde donde colocaremos los paremetros asociados al dato de lectura/escritura via Modbus.

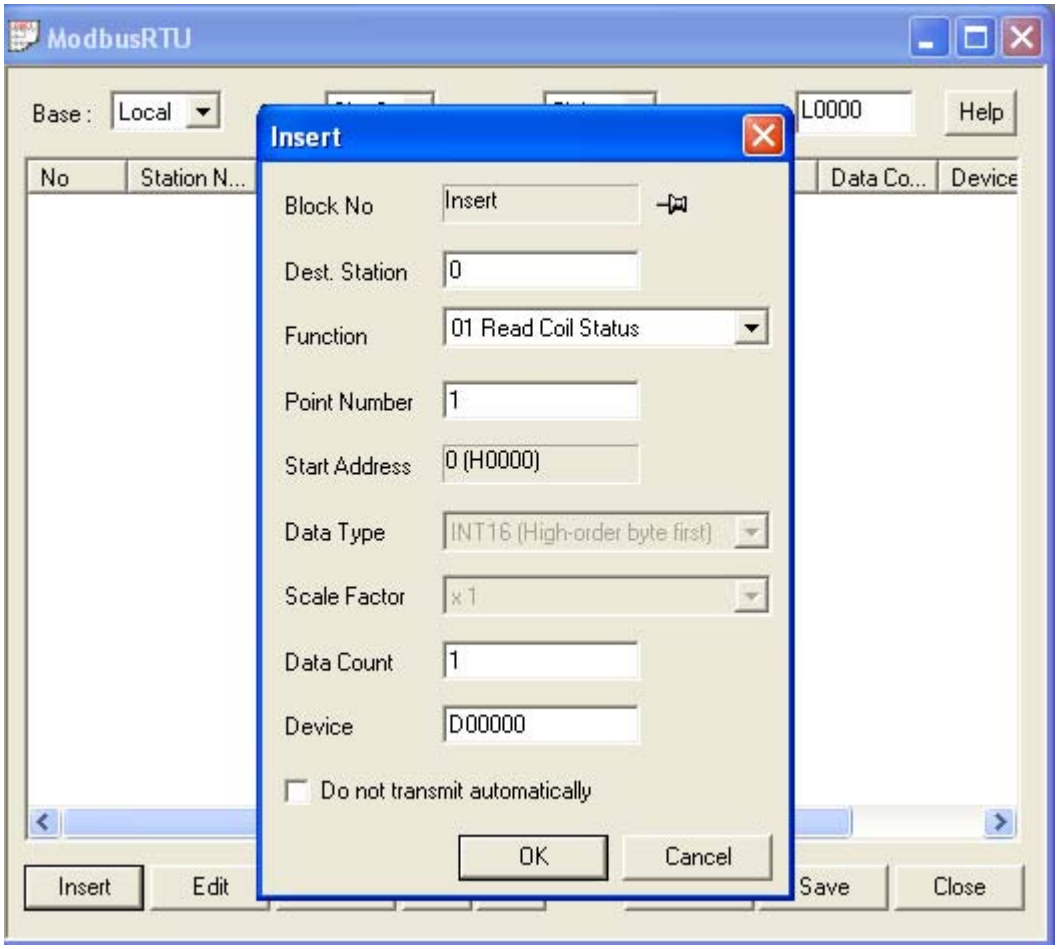


Figura 68: ventana de insercion de parametros Modbus master program.

En el cuadro **Dest. Station** colocamos la dirreccion o numero de estacion del dispositivo u equipo en comunicación via la red Modbus interconectada.

En el cuadro despegable **Function** seleccionaremos la funcion Modbus a utilizar

En el cuadro **Point Number** colocamos la dirreccion de memoria del registro o bit de lectura, este valor siempre es mayor que 0 equivalente a la dirreccion fisica real en la memoria + 1.

En el cuadro Start Address, se podra observar la dirreccion equivalente de memoria relacionada al valor que colocamos en el cuadro Point Number

En el cuadro **Data Type**, colocaremos el tipo de formato de dato que se requiera leer, sea entero, doble entero, float , a si como el orden de bits en la recepcion de los datos.

La opcion data type solo esta disponible si el tipo de variable es diferente de bit.

El cuadro **Scale Factor**, se utiliza cuando se requiera amplificar el tamaño de un numero Float previamente seleccionado, en Data Type.

En el cuadro **Data Count** , se coloca el numero de datos que se leera desde la dirreccion Start Address, según el formato de variables expuesto en Data Type.

En el cuadro **Device,** colocamos la dirreccion de memoria del PLC Cimon, que recepcionara dicho dato, esta dirreccion tiene que ser compatible con el tipo de dato de lectura, es decir, si se lee un bit via modbus, el registro que recepcionara dicha informacion tiene que ser del tipo bit, Ejemplo (Y, K, L, etc)

La opcion “Do not transmit Automatically” permite, desabilitar la transmision de comandos de lectura y escritura via modbus de manera automatica por el programa.

Cuando se habilita esta opcion es necesario usar la instrucción SEND para enviar o transmitir los comandos del bloque de comando seleccionado en el programa. (revisar la forma de operación de la instrucción en el menu de ayuda de Cimon)

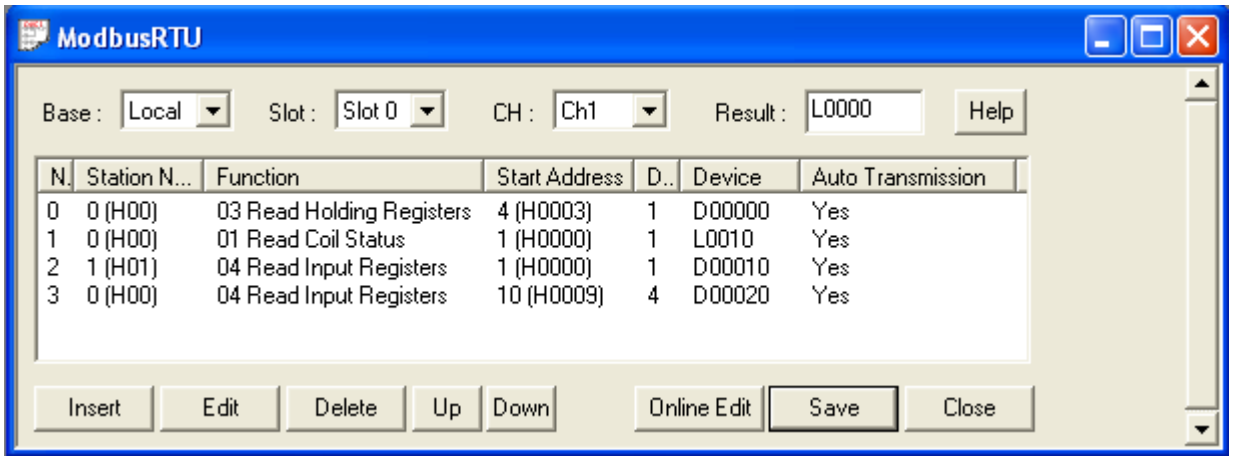

ModbusRTU
Base : Local
Slot : Slot 0
CH : Ch1
Result : L0000
Help
N. Station N... Function Start Address D.. Device Auto Transmission
0 0 (H00) 03 Read Holding Registers 4 (H0003) 1 D00000 Yes
1 0 (H00) 01 Read Coil Status 1 (H0000) 1 L0010 Yes
2 1 (H01) 04 Read Input Registers 1 (H0000) 1 D00010 Yes
3 0 (H00) 04 Read Input Registers 10 (H0009) 4 D00020 Yes
Insert
Edit
Delete
Up
Down
Online Edit
Save
Close

Figura 69: “Communication setting program” para comunicación via modbus RTU

Para poder dirreccionar correctamente los datos de lectura es necesario tener presente la tabla de registros Modbus, del dispositivo esclavo en comunicación.

Con la opcion Slot, mostrada en la figura 69, podemos dirreccionar a algun modulo de expansion de comunicación que este ubicado en dicho slot.

9.2.2 Comunicación via MODBUS TCP modo Maestro.

Al igual que en la comunicación modbus RTU serial, los PLC cimon a traves de su software de programacion CICON, ofrecen la posibilidad de crear plantillas de programa de comunicación para implementar rapidamente la transmision de datos utilizando este protocolo estandar en la industria.

Para agregar un programa nos dirigimos al arbol de proyecto, y agregamos el programa del siguiente tipo:

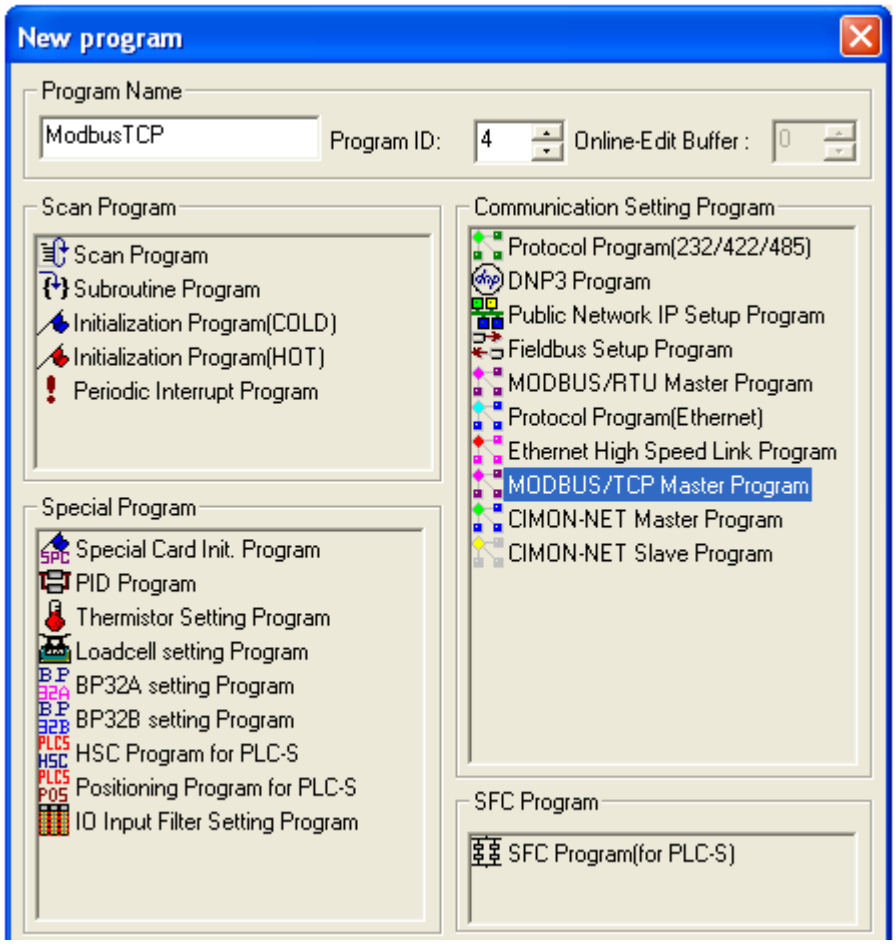


Figura 70: adicion del programa Modbus TCP - program

Una vez agregado el programa podemos comenzar a editar la ventana de configuracion activa que aparece en la siguiente figura:

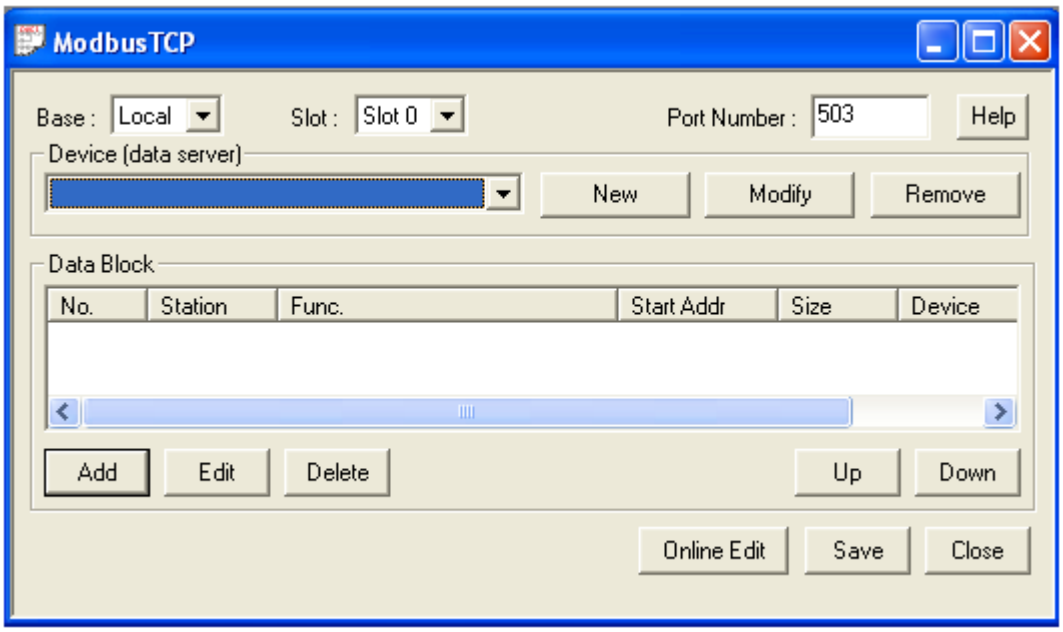
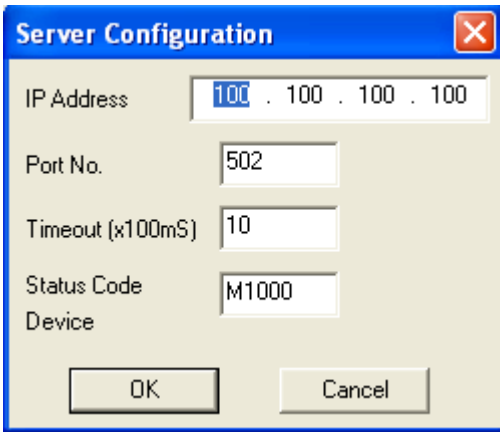


Figura 71: ventana de insercion de parametros Modbus master program.



Al dar clic en New, agregamos la dirreccion IP del dispositivo a comunicar

Figura 72: Ventana para agregar nuevo dispositivo

Es importante destacar que el numero de puerto (socket) por defecto para el protocolo Modbus TCP es 502, sin embargo este puerto puede ser cambiado para personalizar la interfaz de comunicación, lo cual exige que sea tambien declarado en el mismo dispositivo esclavo de la red.

Se debe tener cuidado de fijar el registro que almacene el Estado de comunicación ("Status Code Device") para que no interfiera con su uso dentro de otros programas.

Para agregar un link de solicitud de datos, clic en el boton Add, (ver figura 71).

Una vez hecho esto aparecera la ventana mostrada en la figura 73, aquí se ingresar los parametros necesarios para establecer y enviar una solicitud de datos a travez del protocolo Modbus TCP.

En el cuadro **Station No.,** se coloca el numero de estacion del dispositivo a traves de la red, en caso se haga a travez de varios gateways Ethernet. Este numero debe ser diferente entre los dispositivos esclavos

En el cuadro desplegable **Function,** se selecciona la funcion Modbus en solicitud.

En el cuadro **Start Address,** se coloca el numero de dirreccion del dato a leer sobre el dispositivo esclavo, este valor debe ser obtenido de la tabla de datos Modbus del dispositivo esclavo dentro de la red.

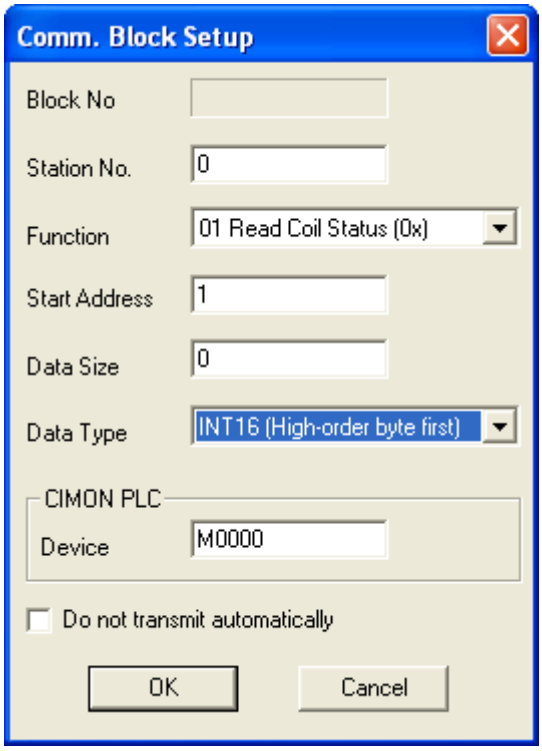


Figura 73: Ventana para agregar nuevo dispositivo

En el cuadro **Data Size**, se coloca el tamaño de datos, a leer desde la dirrecion inicial marcad en el cuadro Start Address.
en el siguiente cuadro se expresa el limite maximo del tamaño de datos a leer/escribir

Command(Function)	Data Size
1 - Read Coil Status	Bit Size (Max 2000 Bit)
2 - Read Input Status	
3 – Read Holding Registers	Word Size (Max 125 Word)
4 – Read Input Registers	
5 – Force Single Coil	Not Necessary.(1Bit Write)
6 – Preset Single Registers	Not Necessary.(2Byte Write)
16 - Preset Multiple Registers	Word Size (Max 100 Word)

En el cuadro **Data Type**, se coloca el tipo de formato de variable que va a ser leida desde el dispositivo esclavo, esta opcion solo tiene implicancia si la funcion modbus es 3,4,6 o 16.

Sobre el cuadro **Device,** se coloca la dirreccion de memoria que almacenara dicha informacion en solicitud según la funcion Modbus implementada. Este formato de variable tiene que ser compatible con la variable de retorno en solicitud.

La opcion “Do not transmit Automatically” permite, desabilitar la transmision de comandos de lectura y escritura via modbus de manera automatica.

Luego de haber ingresado las dirrecciones de lectura la ventana de interfaz de programacion para modbus TCP, podra aparecer de la siguiente manera.

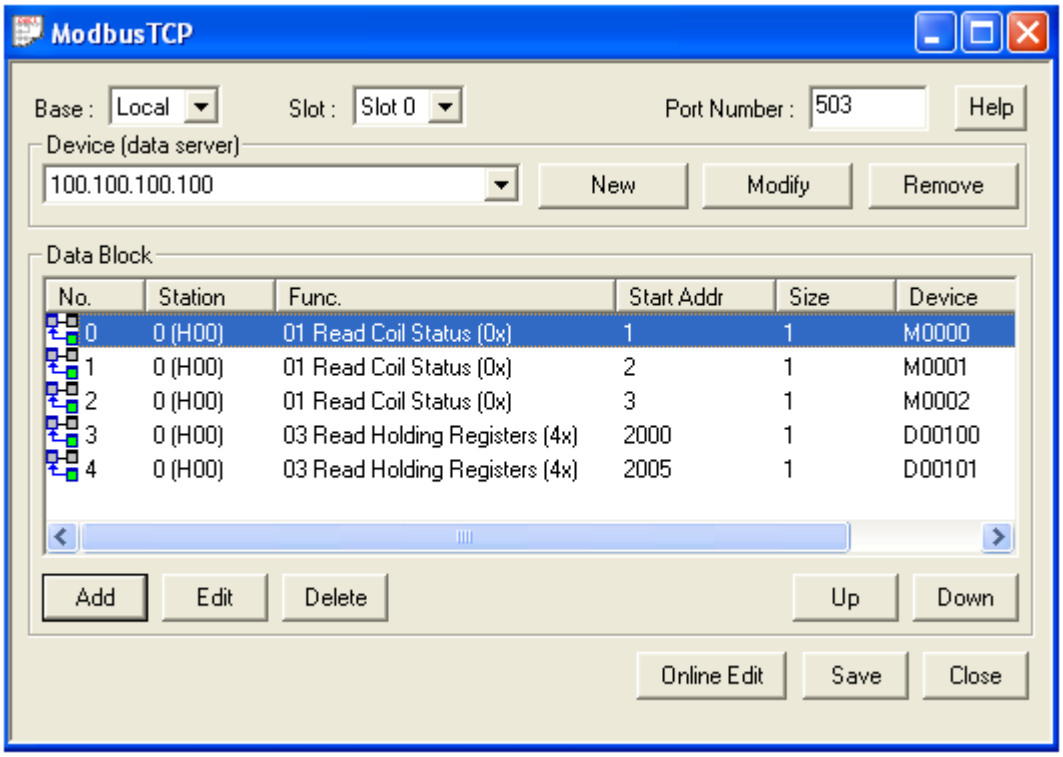


Figura 74: “Communication setting program” para comunicación via modbus TCP

En esta ventana tambien se podra establecer el slot respectivo desde donde se encuentra la interfaz ethernet que permitira la comunicación via el protocolo Modbus TCP.

9.3 Creacion de un protocolo personalizado.

En el PLC cimon es posible implementar un protocolo propio en funcion de comandos o tramas de datos que se transmita y capture en su recepcion.

Esto puede muy util cuando se requiera comunicar con un equipo remoto, a travez de comandos.
De esta forma podremos habilitar la comunicación sin necesidad de establecer codigo de programa dentro del programa principal, si no a travez de una interfaz de configuracion para administracion y filtrado de datos.

Para crear un programa protocolo, es necesario crear un nuevo programa y seleccionar el tipo mostrado en la figura siguiente:

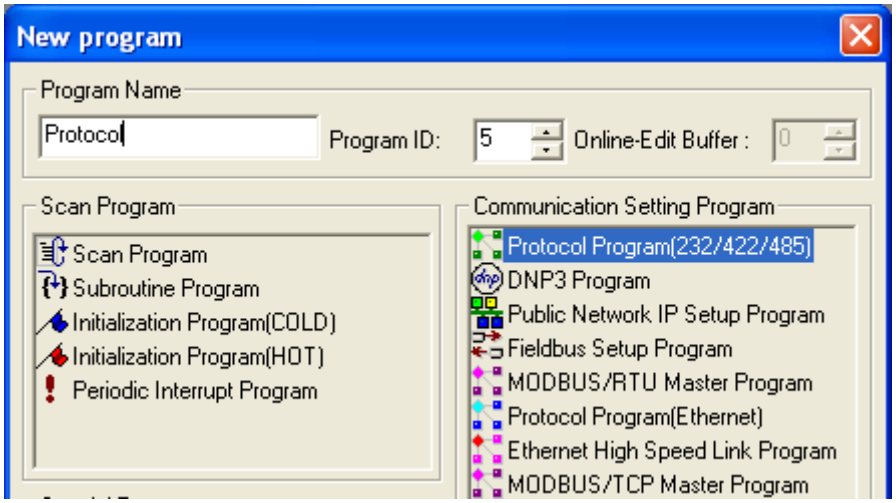


Figura 75: Creacion de programa Protocolo.

Existe dos medios para crear los programas protocolo, esto via Serial, o via el puerto Ethernet, en ambos casos las opciones de implementacion estan habilitadas.

Una vez creado, tenemos que tener en cuenta cual es el algoritmo que consta el protocolo que deseamos implementar.

Al crear un programa Protocolo aparecera la ventana mostrada en la figura 76, una vez ahí nosotros podemos agregar cualquier rutina de tramas de comunicación, con el boton Add. Esta trama podra transmitir o recepcionar algun dato requerido o filtrar el recibido mediante las opciones que presentra este editor.

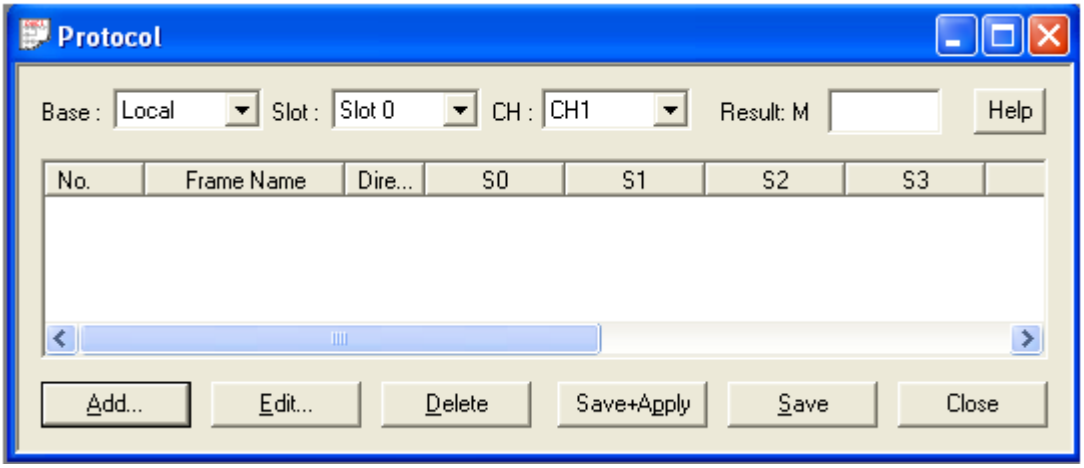


Figura 76: Interfaz del editor de programa Protocolo.

9.3.1 Ventana de Edicion y creacion de tramas

En la siguiente ventana se muestra la interfaz donde nosotros podremos agregar las tramas que constaran nuestro protocolo de comunicación.

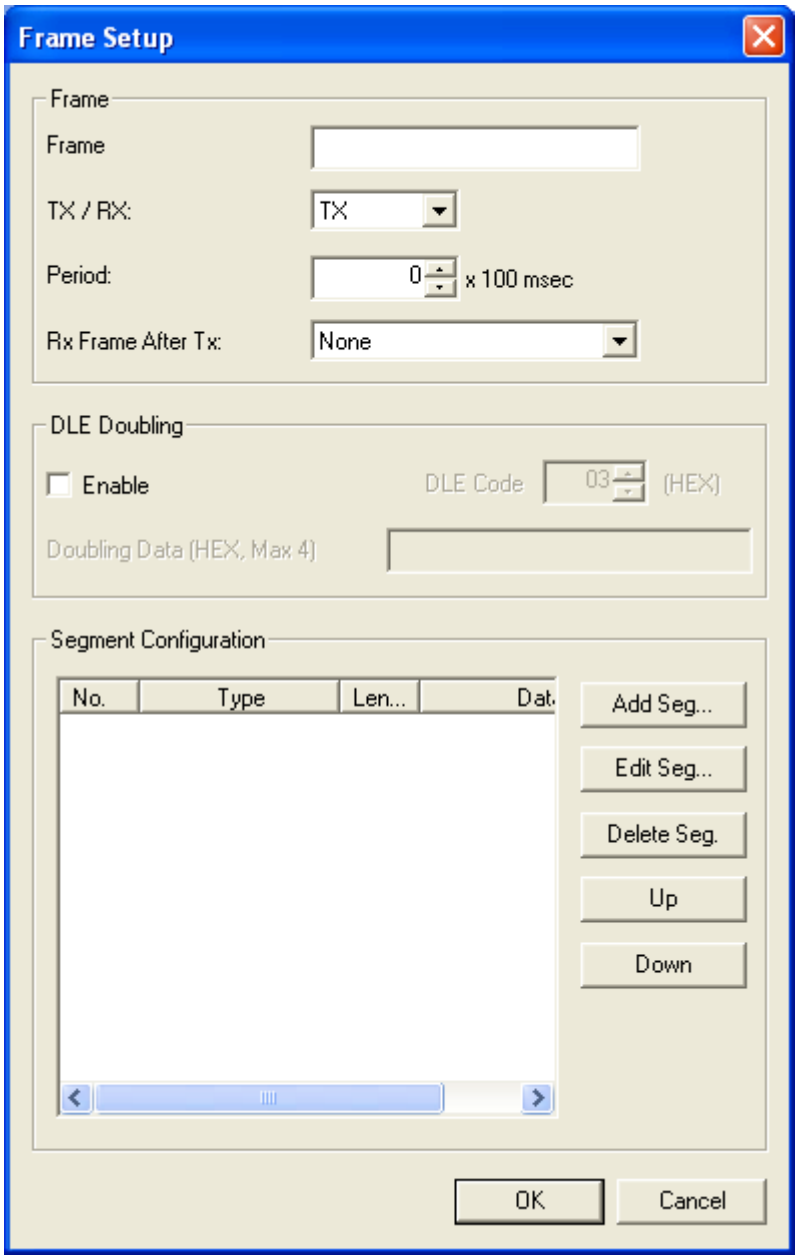


Figura 77: Interfaz del editor de tramas.

En la siguiente ventana se muestra la interfaz donde nosotros podremos agregar las tramas que constaran nuestro protocolo de comunicación.

Donde:

Frame, en este campo se coloca el nombre de la trama a crear.
TX/RX, aquí podemos seleccionar la opcion de transmitir o recepcionar los datos.
Period, indicamos el periodo con que la trama se repetira automaticamente.
Rx Frame After Tx, especificamos el nombre de trama que se ejecutara como respuesta.

Existen funciones relacionadas con la edicion de los segmentos:

Add Seg, define y agrega un nuevo segmento
Edit Seg, edita detalles del segmento registrado
Delete Seg, elimina el segmento seleccionado
Up, se mueve sobre el segmento superior
Down, se mueve sobre el segmento inferior

9.3.2 Creacion de tramas de Transmision / recepcion de datos

En la ventana de interfaz de editor de tramas, (figura 77) presionamos el boton “Add Seg..”

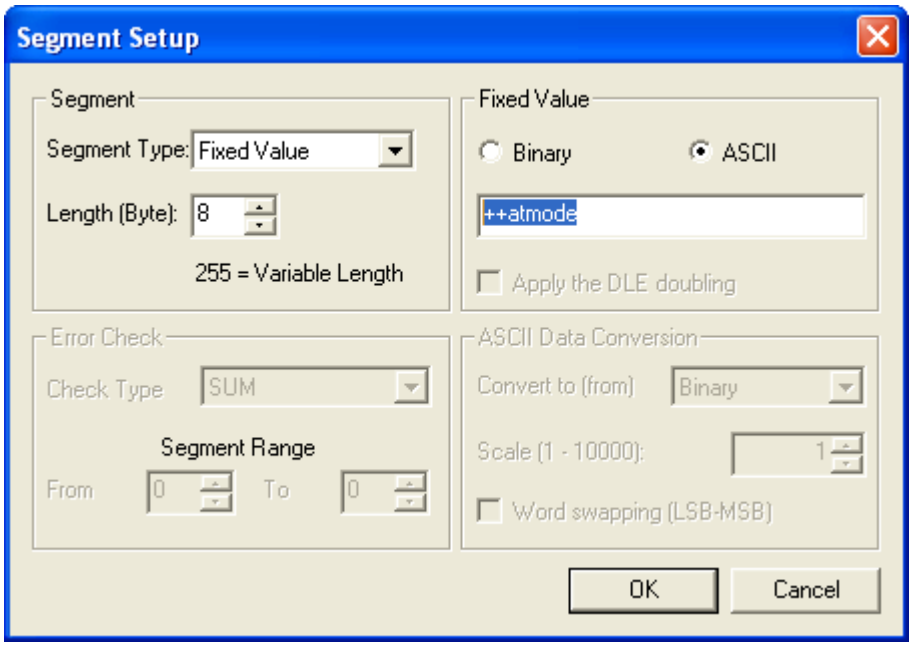


Figura 78: Ventana de edicion de segmento

En esta nueva ventana seleccionaremos el formato de tipo de dato a transmitir, Binario o ASCII, Si seleccionamos ASCII, podemos ingresar en el cuadro de texto, los caracteres a transmitir. En la figura anterior mostrada se ingresa “++atmode” que sera un codigo de comando para comunicarse con un dispositivo.

Tambien seleccionamos el tipo de segmento, que puede ser de:

- Fixed Value, con un valor constante de datos tanto en la transmision o recepcion.
- Ignore, en la cual los datos a recepcionar pueden ser ignorados en un numero determinados de bytes (Length Byte).
- Memory link, el cual almacenara o tomara datos desde un registro de memoria del PLC.
- Error Check (BCC), permite implementar la identificacion de errores en la comunicacion

Entonces para nuestro primer comando de transmision, seleccionamos el tipo caracteres fijos, clic en OK.

Luego de regreso en la ventana del editor de tramas colocaremos como opcion Transmitir (TX) con periodo de 100ms, ademas del nombre para esta primera trama de datos “inicial”.

luego de esto en nuestra ventana principal de creacion de protocolo (figura 76) aparecera un nuevo item de trama de datos:

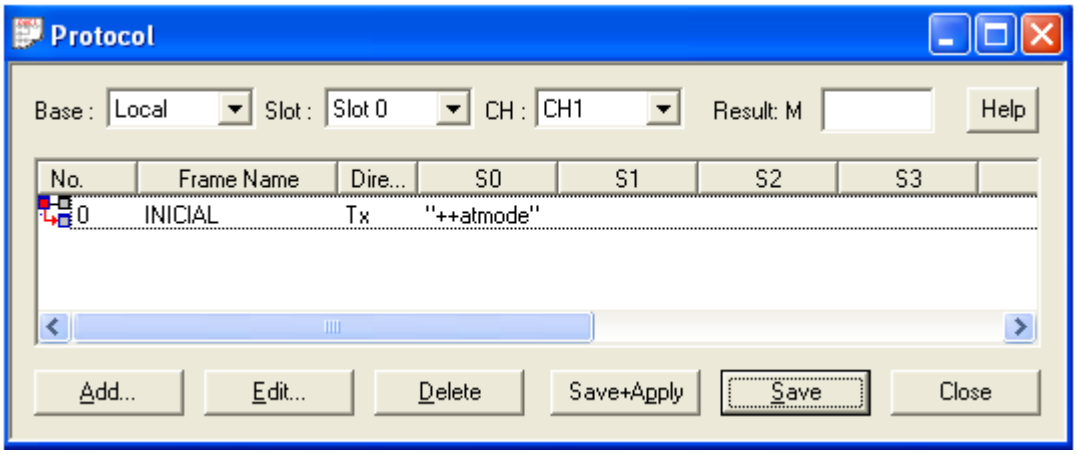


Figura 79: ventana de creacion de protocolo con trama de transmision

Luego de esto tambien podemos agregar un nueva trama en modo recepcion.

Presionamos el boton Add sobre la ventana principal de creacion de protocolo y esta vez creamos una trama en modo recepcion.

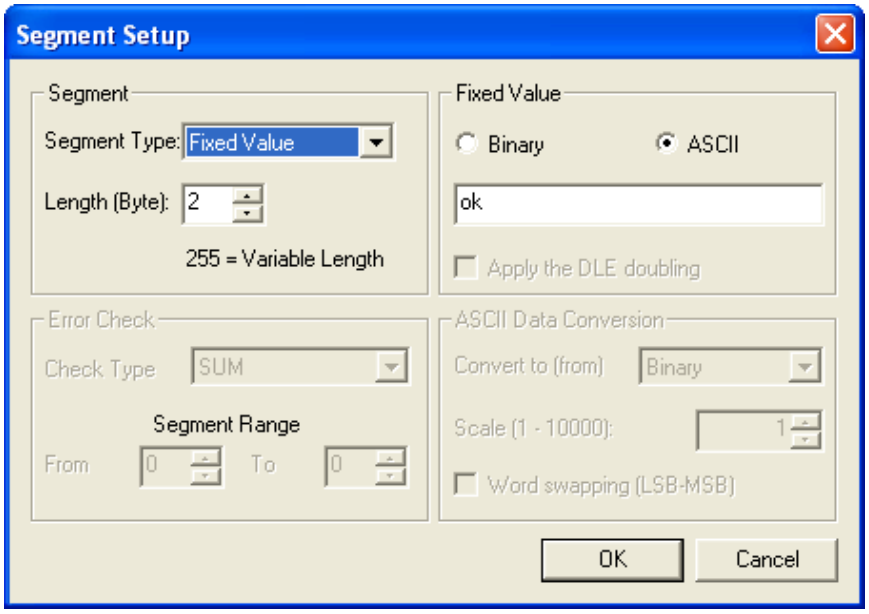


Figura 80: ventana de edicion de segmento con dato fijo esperado de respuesta.

Colocamos el texto “OK, que sera la respuesta esperada desde nuestro dispositivo, una vez que tramisitimos el primer comando “++atmode”.

para poder relacionar nuestra respuesta esperada con la transmision de datos, es necesario agregar sobre las propiedades de la primera trama , la opcion Respuesta sobre transmision.

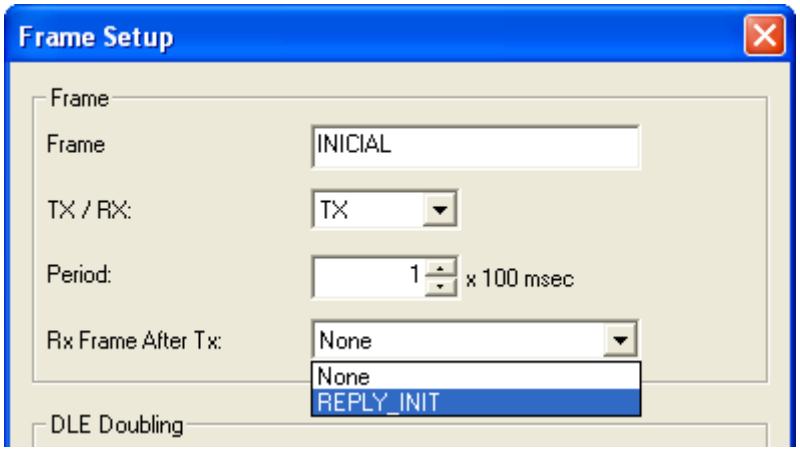


Figura 81: modificacion de trama de salida, con respuesta relacionada.

en la figura anterior se muestra que para la trama “INICIAL”, se agrega como respuesta esperada “REPLY_INIT”.

Para poder seguir enlazando la siguiente ruta de tramas creamos, nuevas tramas de transmision y respuesta.

- Una trama transmision con datos fijos, pero 2 segmentos de datos
- Una trama de respuesta que guarde los datos en un registro del PLC.

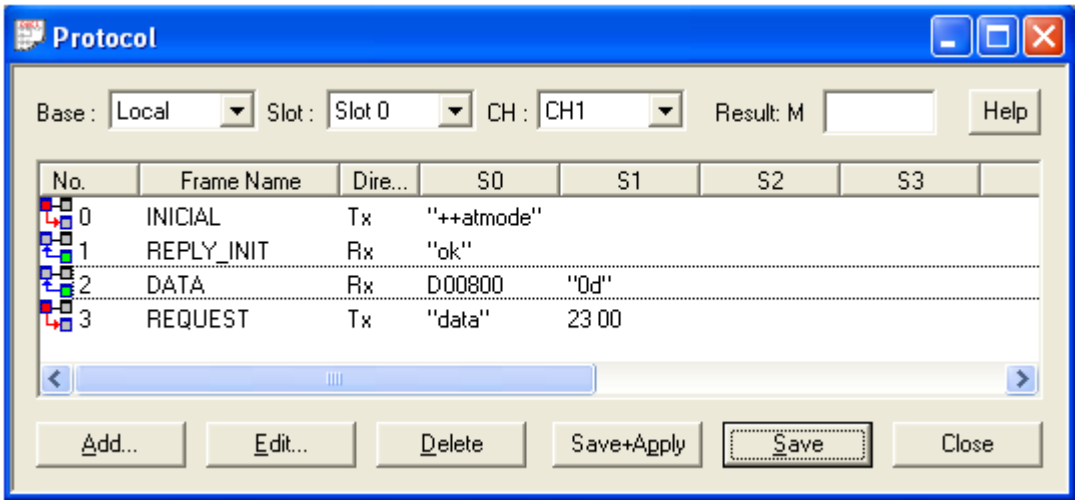


Figura 82: Ventana protocol con 4 tramas de datos.

Dentro de la nueva trama de transmision de datos, colocamos 2 segmentos, uno con el comando “data”, con 2 caracteres binarios 23,00.

Esto podria hacerse analogia con la solicitud de lectura del registro ubicado en la dirreccion 0x2300, del dispositivo.

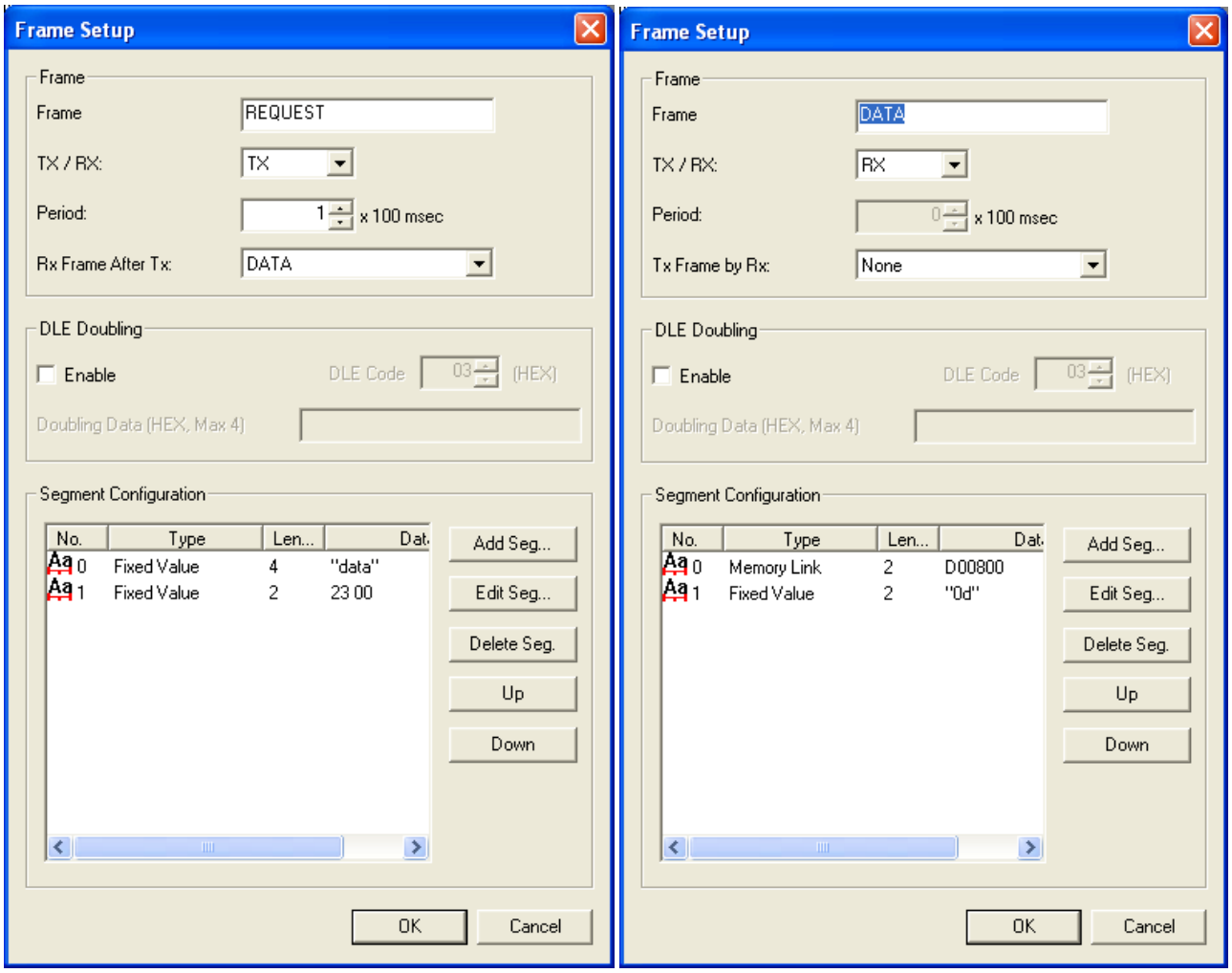


Figura 83: Ventanas de configuracion de tramas transmision y recepcion.

En este ejemplo se crea otra trama de recepcion la cual almacena el dato recibido en el registro D00800, y se espera como dato final el caracteres “0d”. Para guardar el dato recibido en un registro debemos utilizar la opcion Memory link mostrado en la figura siguiente:

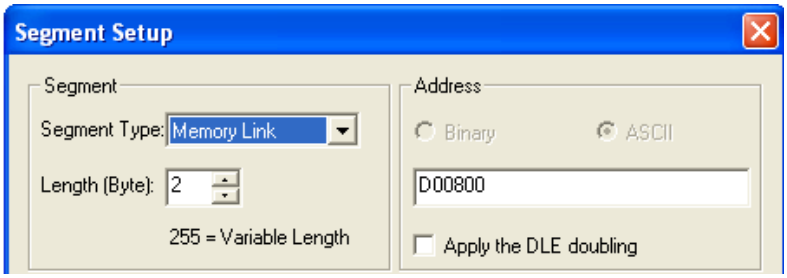


Figura 84: Ventanas de configuracion de segmento para guardar dato sobre registro.

De esta forma nuestra flujo de datos en comunicación con el dispositivo se vera de la siguiente manera:

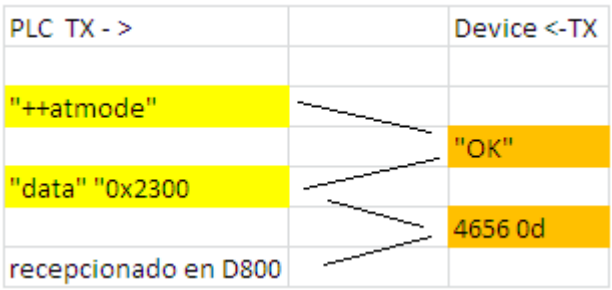


Figura 85: Flujo de datos implementado en programa protocolo.

En este ejemplo el dispositivo en comunicación, envia la palabra “OK” cuando recibe el comando “++atmode”, y devuelve el valor de 4656 como valor del registro de memoria 0x2300 solicitado por el plc utilizando los comandos data.

Este valor es almacenado en el registro del PLC D800, gracias a la implementacion de tramas de respuesta.

No se debe olvidar indexar los respuestas o transmisiones de datos en funcion de una trama anterior cuando fuese necesario hacerlo, (ver figura figura 81).

10. FUNCIONES ESPECIALES INCORPORADAS

10.1 BLOQUE CONTROL PID CON EL PLC CIMON.

Dentro del PLC Cimon, nosotros podemos implementar un bloque de programa con controles y bucles PID de manera rapida, esto lo podemos hacer agregando un nuevo programa a nuestro proyecto:

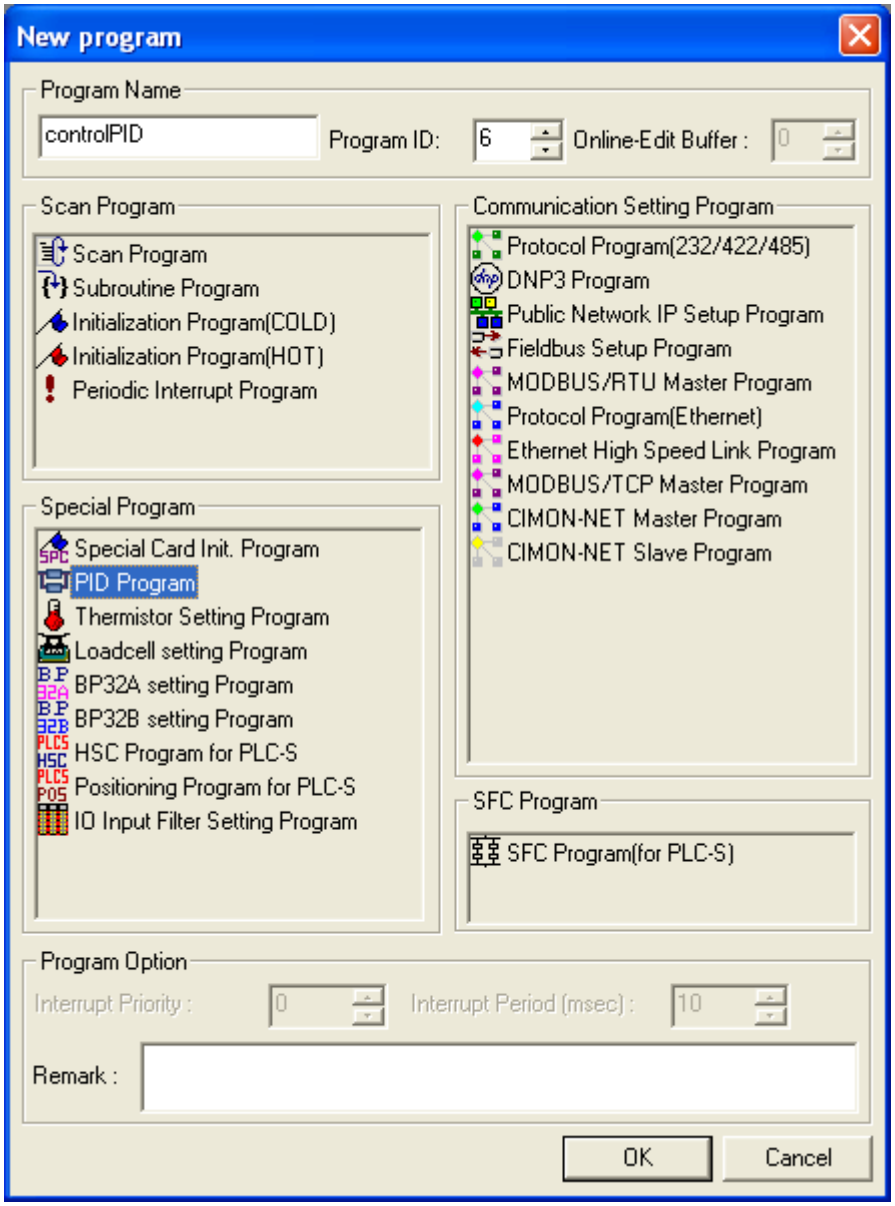


Figura 86: Adicion de un bloque de programa especial PID.

Cuando se agrega un nuevo programa se mostrara el cuadro mostrado en la figura 87, Para poder entender mejor el manejo de este programa vamos a explicar los parametros mas importantes dentro del control PID.

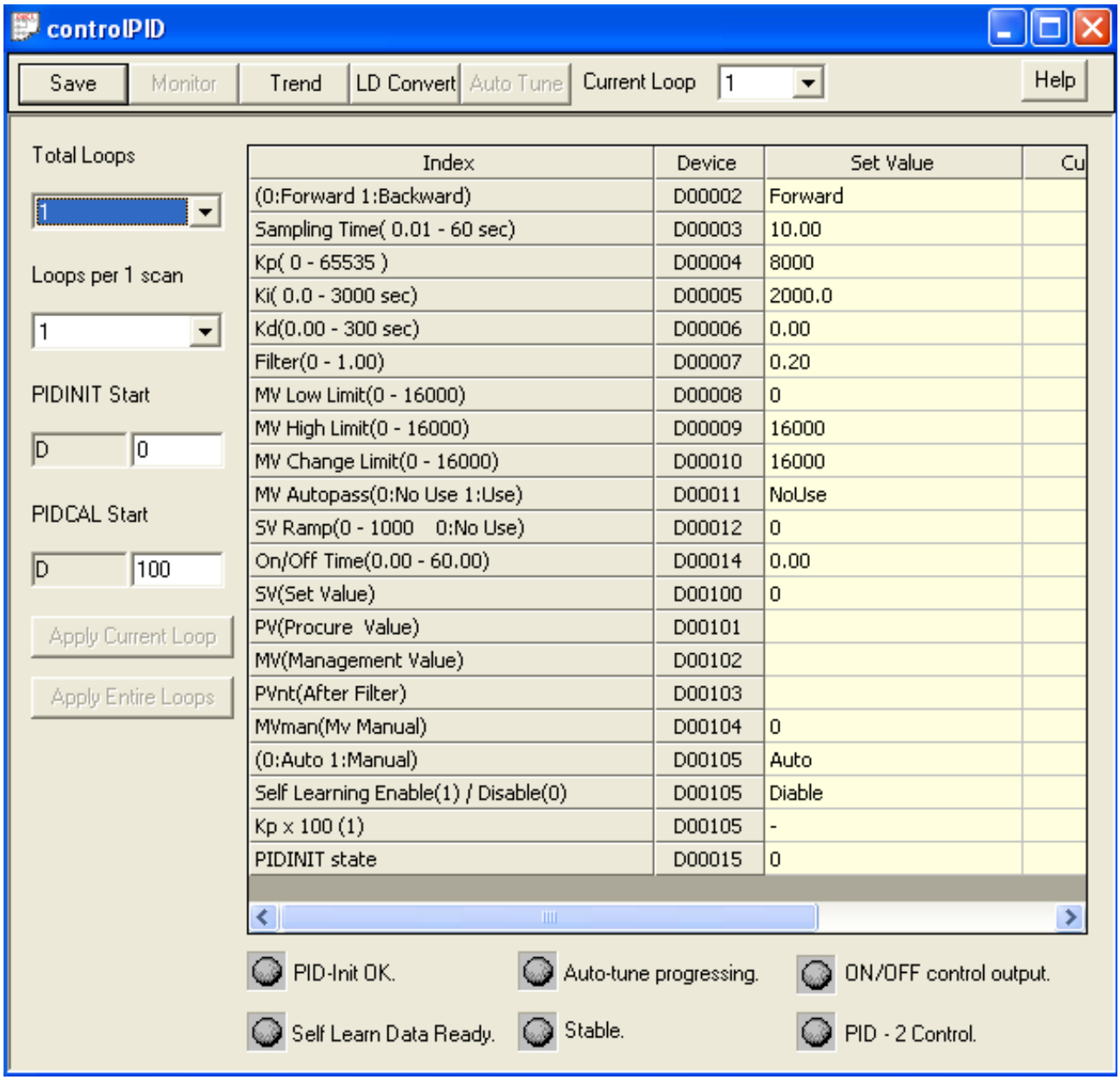


Figura 87: Bloque de programacion PID.

10.2 Parametros Basicos dentro del control PID.

Dentro del PLC Cimon, nosotros podemos implementar un bloque de programa con

Process variable (PV), señal de entrada que va a ser monitoreada. Puede ser el registro que almacena el valor actual de medida del sensor.
Manipulation variable (MV), Es la señal de salida que efectuara el control sobre el proceso, esta señal va hacia al actuador y estará apuntando al registro de salida.
Set value (SV), Es el valor de Set point ajustado por el controlador del sistema o usuario que requiere que dicha variable en control permanezca en un valor determinado.
Error, Es el resultado de la diferencia entre SV-PV.

The PID control process showing as above configuration receives the process variable (PV) and controls the manipulation variable (MV) in order to adjust the PV to match the set value (SV). D/A conversion module is used for MV output.
Es muy posible que los valores de salida de la variable MV pasen directamente a un canal del conversor D/A.

El primer parámetro a configurar mostrado en la figura 87, es el modo de operación el cual puede aplicarse modo directo/inverso:

Direct action: El valor de MV se incrementa cuando PV es mayor que SV
Reverse action: El valor de MV se decrementara cuando PV es mayor que SV.

Proportional integer (Kp)
Esta es la constante que multiplicara el valor de desviasion (Sv –Pv) en la operacion proporcional "**Mv = Kp * (Sv – Pv)**". Este valor puede ser ajustado des de 0.0 a 100.

Integral integer (Ki)
Es la constante de ganancia integral comunmente usado para remover el offset de salida final, el valor puede ser ajustado de (0.0 ~ 3000).

Differential integer (Kd)
Es la constant de ganancia derivative, proporcional a los cambios rapidos del sistema y las desviaciones precentes que mediante las ganacias Integral y proporcional no pueden ser corregidas, este valor puede ser ajustado en el rango de (0.00 ~ 300).

Automatic MV transfer (0: Use, 1: No use)
Se utiliza para actualizar la salida MV manual, igual a la salida del modo automatico cuando el modo de ejecución se encuentra en modo automatico.
Si no se usa esta opción, el valor ajustado en MVman será aplicado hacia MV en modo manual.

SV ramp (0: No use, 1~1000)
Genera la rampa en el cambio del set point de control

ON/OFF control time(On/Off time)
Esta opción permite controlar el nivel de salida en modo de pulsos, con el periodo predefinido en el valor ON/OFF time. Se deshabilita esta opción si se pone a 0.
El tiempo en que permanece encendido viene definido en el valor actual de la salida MV:
Ton = TimeOnOff * MVactual / MVmaxn

10.3 Formula de salida para el control PID en funcion de sus parametros

[Forward Action]

- $MV_n = MV_{n-1} + K_p\{(E_n - E_{n-1}) + (T_s/K_i)*E_n + (K_d/T_s)*(2PV_{nf-1} - PV_{nf} - PV_{nf-2})\}$
- $E_n = SV - PV_{nf}$

[Reverse Action]

- $MV_n = MV_{n-1} + K_p\{(E_n - E_{n-1}) + (T_s/K_i)*E_n - (K_d/T_s)*(2PV_{nf-1} - PV_{nf} - PV_{nf-2})\}$
- $E_n = PV_{nf} - SV$

[Filtered Present Value]

- $PV_{nf} = PV_n + á(PV_{nf-1} - PV_n)$

P I D

$$MV_n = MV_{n-1} + K_p\{(E_n - E_{n-1}) + (T_s/K_i)*E_n + (K_d/T_s)*(2PV_{nf-1} - PV_{nf} - PV_{nf-2})\}$$

En : Currently Sampling Deviation
En-1 : Deviation at an Interval before
Kp : Propotional Integer
Ki : Integral Integer
Kd : Differential Integer
Ts : Sampling Interval
¥á : Filter Coefficient
MVn : Present manipulation value
MVn-1 : Manipulation value of the preceding sampling cycle
SV : Set Value
PVn : Process value of the present sampling cycle.
PVnf : Process value of the present sampling cycle (after filtering).
PVnf-1 : Process value of the preceding sampling cycle (after filtering).
PVnf-2 : Process value of the sampling cycle two cycles before (after filtering).

10.4 Rango de valores maximos para parametros PID

Item		Specification	Remarks
Range to Set PID Constant	Proportional Constant(K_p)	1 – 10000 (0.01 – 100.00)	
	Integral Constant (K_i)	0 – 30000 (0.0s – 3000.0s)	0 : No Integral Operation
	Differential Constant (K_d)	0 – 30000 (0.00s – 300.00s)	0 : No Differential Operation
Sampling Interval (T_s)		1 – 6000 (0.01s – 60.00s)	
Range of Set Value(SV)		0 – 16,000	
Range of Current Value(PV)		0 – 16,000	
Range of Manipulated Output Value(MV)		0 – 16,000	
Range of Manually Manipulated Value		0 – 16,000	
PID Control Loop		32 Loop	
Control Operation		Normal/Reverse Operation	
Type of Operation		Differentiates measured values	

10.5 Distribucion de registros de memoria en un Bloque de control PID

El bloque de programa PID, direcciona registros de configuración de manera automática, (ver figura 87), sin embargo la posición de estos registros puede ser personalizada, así como la conversión de todo este bloque a un código de programa equivalente en ladder, que cumple la misma función de control PID.

Es importante saber separar el tamaño de memoria disponible para el bloque de control PID, ya que la mala distribución de esta podría ocasionar que los datos se guarden correctamente en su ubicación.

Este bloque de control ofrece también las ventajas de poder crear Bucles dentro de un mismo control PID. Sin embargo se debe tener en cuenta que el tamaño de de memoria comienza a aumentar y es importante saber direccionar correctamente los registros.

10.5.1 PIDINIT starting area (PIDINIT Start)

Estar área está reservada para los parámetros superiores dentro del bloque de programa PID. El número total de registros para el primer bucle es 22 (Word), para el Segundo bucle el area reservada es de 20 registros (Word)

En el grafico se siguiente (figura 88) se observa que el espacio de memoria para el siguiente bucle se encuentra a partir del registro D22, aun cuando la posición de memoria Inicial del primer bucle comenzó en 0. Esto demuestra que el minimo de registros para la configuración del pidinit es de 22 (D0 – D21).

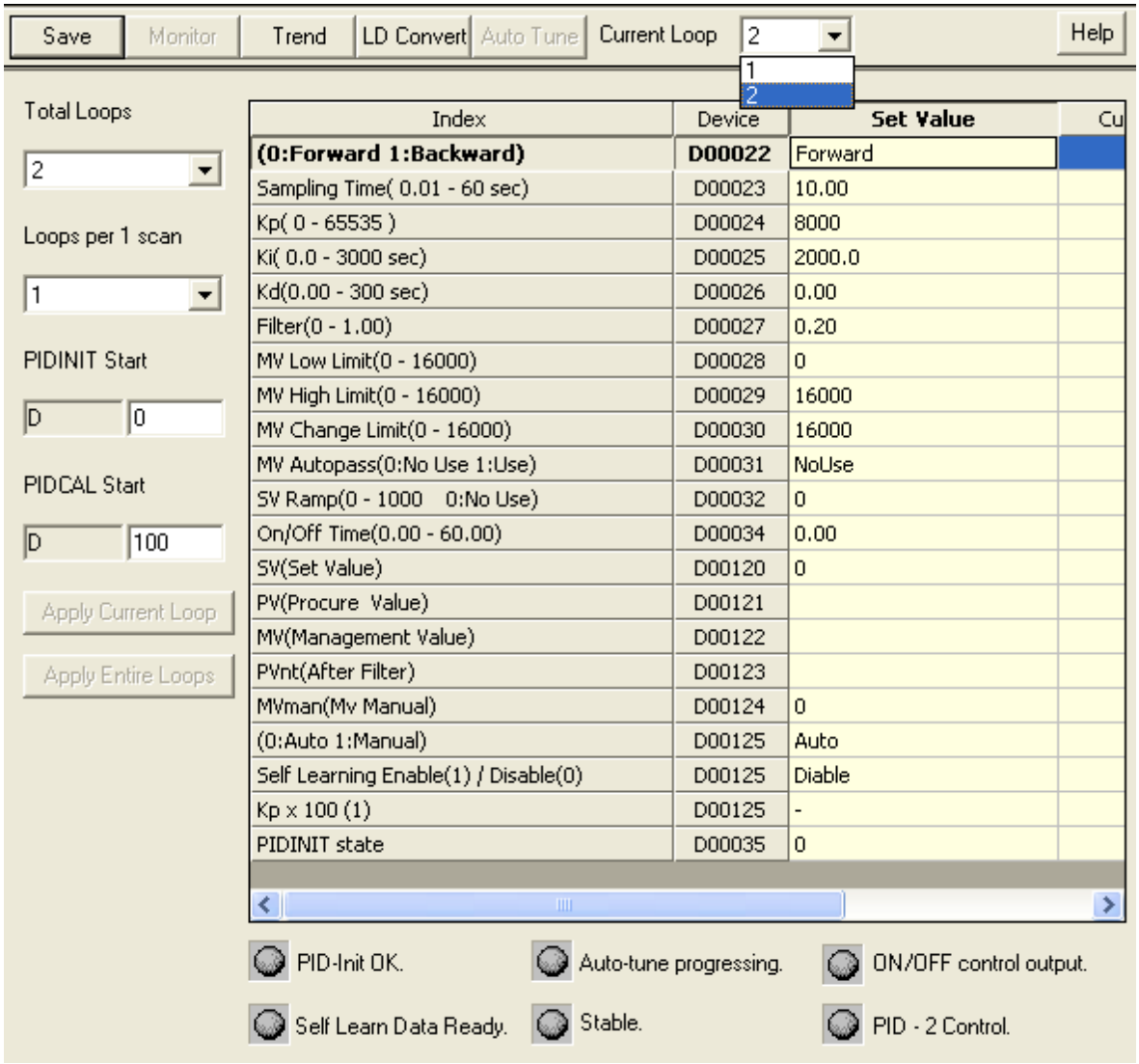


Figura 88: Bloque de programacion PID.

10.5.2 PIDCAL starting area(PIDCAL Start)

Aqui se encuentran los parametros inferiores del bloque de programa PID, el area reservada minima para estos parametros es 20 Words, es necesario ser cuidadoso cuando se requiera usar registros dentro de esta area reservada, ya que es muy comun que se utilize los registros del area inicial (PIDinit start).

10.6 Graficas de tendencias con control PID

Es posible realizar graficas de tendencias previamente ya configurados presionando el botón trend sobre el bloque de programa PID.

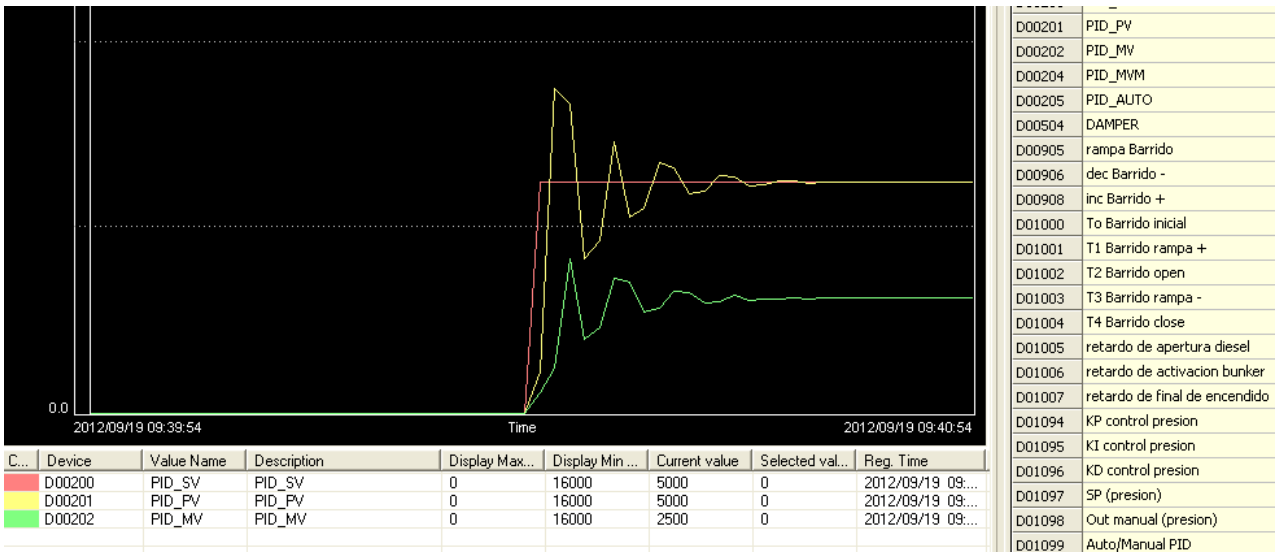

0.0
2012/09/19 09:39:54
Time
2012/09/19 09:40:54
C... Device Value Name Description Display Max... Display Min ... Current value Selected val... Reg. Time
D00200 PID_SV PID_SV 0 16000 5000 0 2012/09/19 09:...
D00201 PID_PV PID_PV 0 16000 5000 0 2012/09/19 09:...
D00202 PID_MV PID_MV 0 16000 2500 0 2012/09/19 09:...
D00201 PID_PV
D00202 PID_MV
D00204 PID_MVM
D00205 PID_AUTO
D00504 DAMPER
D00905 rampa Barrido
D00906 dec Barrido -
D00908 inc Barrido +
D01000 To Barrido inicial
D01001 T1 Barrido rampa +
D01002 T2 Barrido open
D01003 T3 Barrido rampa -
D01004 T4 Barrido close
D01005 retardo de apertura diesel
D01006 retardo de activacion bunker
D01007 retardo de final de encendido
D01094 KP control presion
D01095 KI control presion
D01096 KD control presion
D01097 SP (presion)
D01098 Out manual (presion)
D01099 Auto/Manual PID

Figura 89: Grafica de tendencia sobre bloque de programa PID.

- Es importante destacar que el El bloque PID se ejecuta automáticamente añadiéndose como un ‘scan program’, por eso para habilitar o inhabilitar su funcionamiento, se debe manipular los parámetros de control, para hacer que su efecto sea nulo sobre la salida.
- Para convertir este bloque de programa PID, a su equivalente en Ladder se puede presionar el botón LD convert, dentro de la pantalla principal de programa (figura 87)

11. CONEXIONES ELECTRICAS EN LOS PLCs

Estar área está reservada para los parámetros superiores dentro del bloque de programa PID. El número total de registros para el primer bucle es 22 (Word), para el Segundo bucle el área reservada es de 20 registros (Word).

En el grafico se siguiente (figura 88) se observa que el espacio de memoria para el siguiente bucle se encuentra a partir del registro D22, aun cuando la posición de memoria Inicial del primer bucle comenzó en 0. Esto demuestra que el mínimo de registros para la configuración del pidinit es de 22 (D0 – D21).

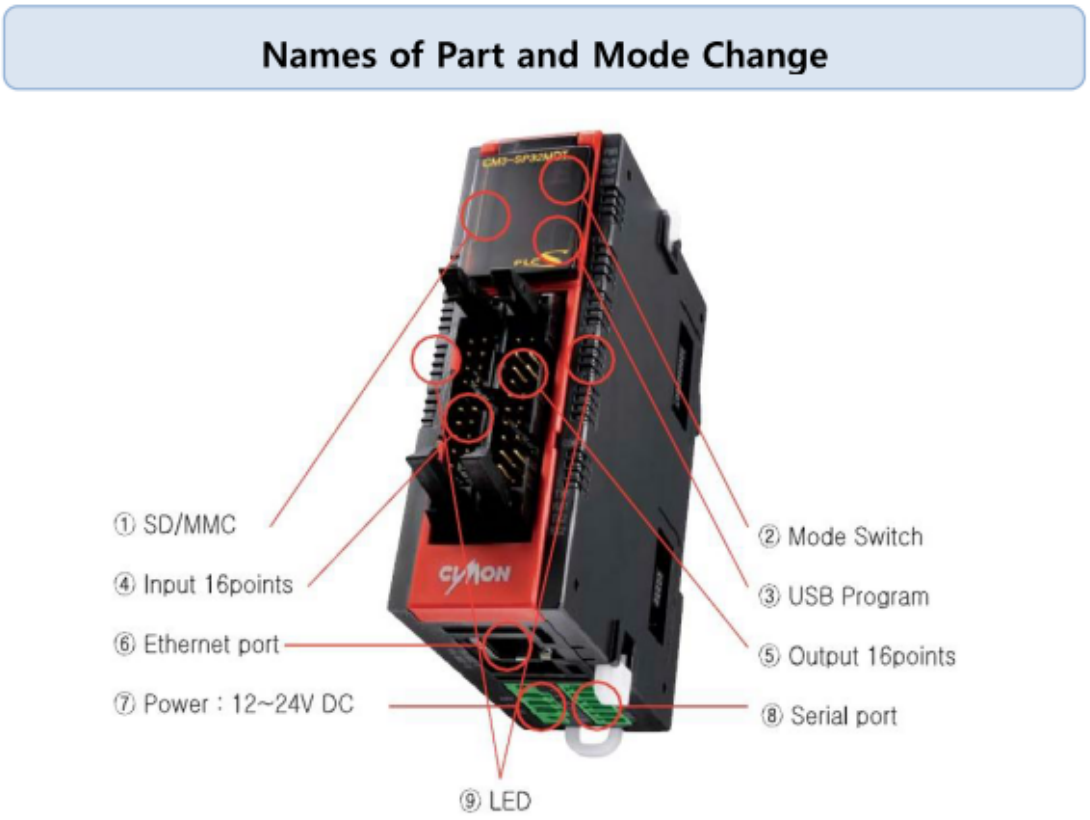


Figura 90: Partes y componentes PLCs CM3

Los PLC serie S, se caracterizan por ser bastante compactos y presentar los puertos básicos de comunicación disponibles, ETHERNET, SERIAL RS485-RS232, USB,

La tensión de alimentación de este PLC tiene que ser 24VDC,
La alimentación de los módulos de expansión se realiza a través del PLC, exceptuando los módulos especiales tales como los módulos Análogos, que requieren alimentación externa para generar sus señales de salida, y darle la potencia suficiente.

11.1 Descripcion de puertos fisicos de comunicación.

11.1.1 Conexión del puerto serial:

Existe 2 formas de comunicarse vía serie, utilizando la interfaz RS232 y RS485, los terminales de los puertos serie del PLCs estan etiquetadas en el mismo equipo.

Para establecer una comunicación en linea con una PC, es necesario armar un cable serie con un terminal hembra (DB9 Female).
Y luego si el cable no es muy largo, usar cualquier cable serie directo.

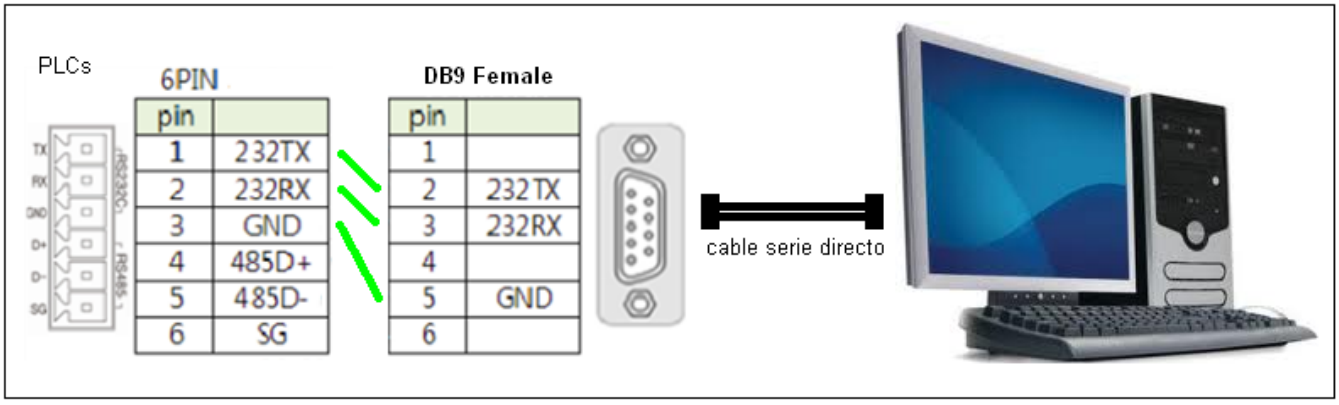


Figura 91: conexión de PC con cable serie RS232.

Cuando se requiera conectar con otro dispositivo usando el puerto serie RS232, es necesario tener en cuenta que los terminales Tx y Rx deben conectarse a sus contrapuestos Rx y Tx del otro dispositivo.

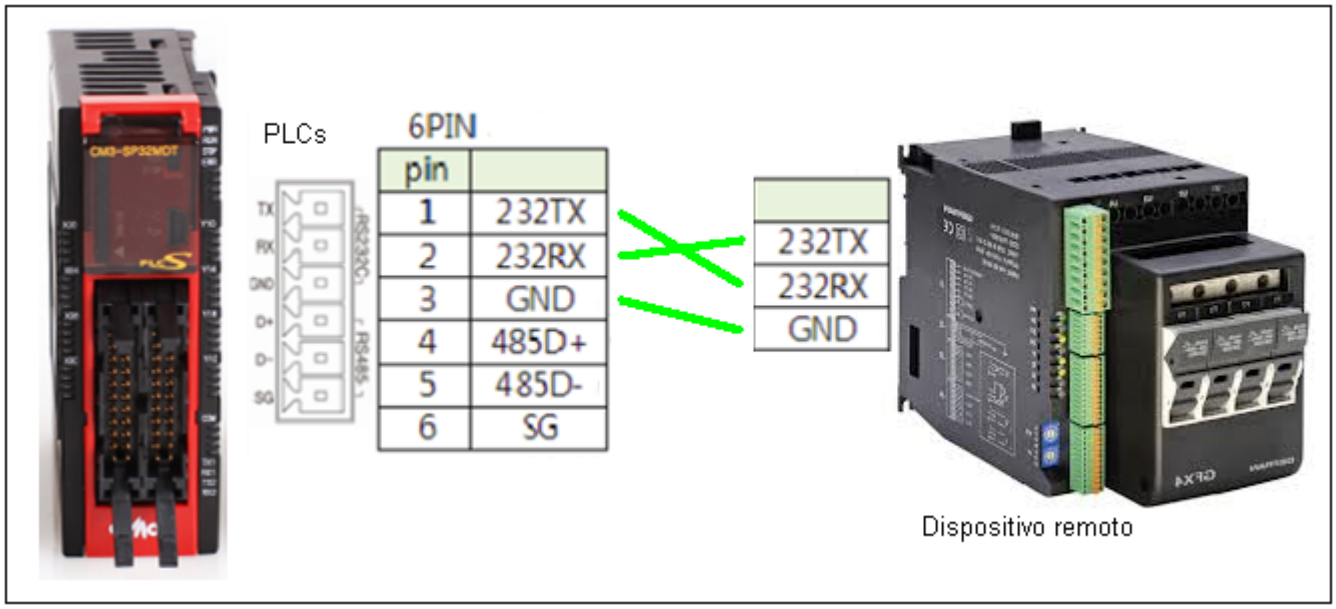


Figura 92: conexión entre PLCs y Dispositivo remoto via Rs232.

Cuando se requiera conectar con otro dispositivo usando el puerto serie RS232, es necesario tener en cuenta que los terminales Tx y Rx deben conectarse a sus contrapuestos Rx y Tx del otro dispositivo. Mediante una conexión en serie no se pueden conectar equipos en red, ya que la comunicación es solo punto a punto.

Cuando se requiera usar la conexión de una red con equipos en paralelo, se utiliza el puerto RS485.

- Si en caso no es posible utilizar el puerto RS485, y se necesite realizar la comunicación en red con varios equipos, se pueden utilizar convertidores RS232 a RS485, o dispositivos inalambricos (modem RF, Zigbee, etc) que establezcan una red externa fuera del puerto RS232 del PLCs.

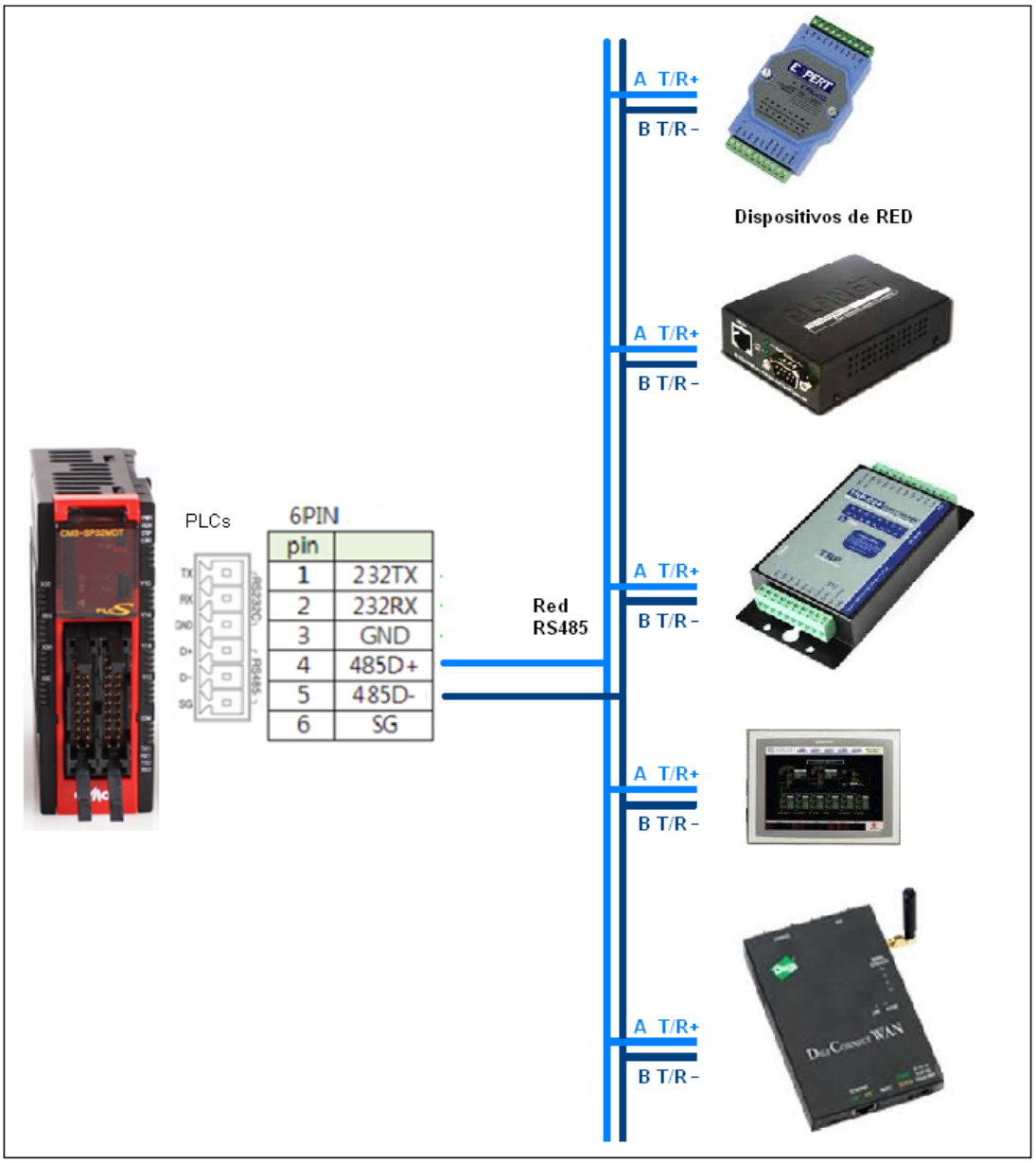
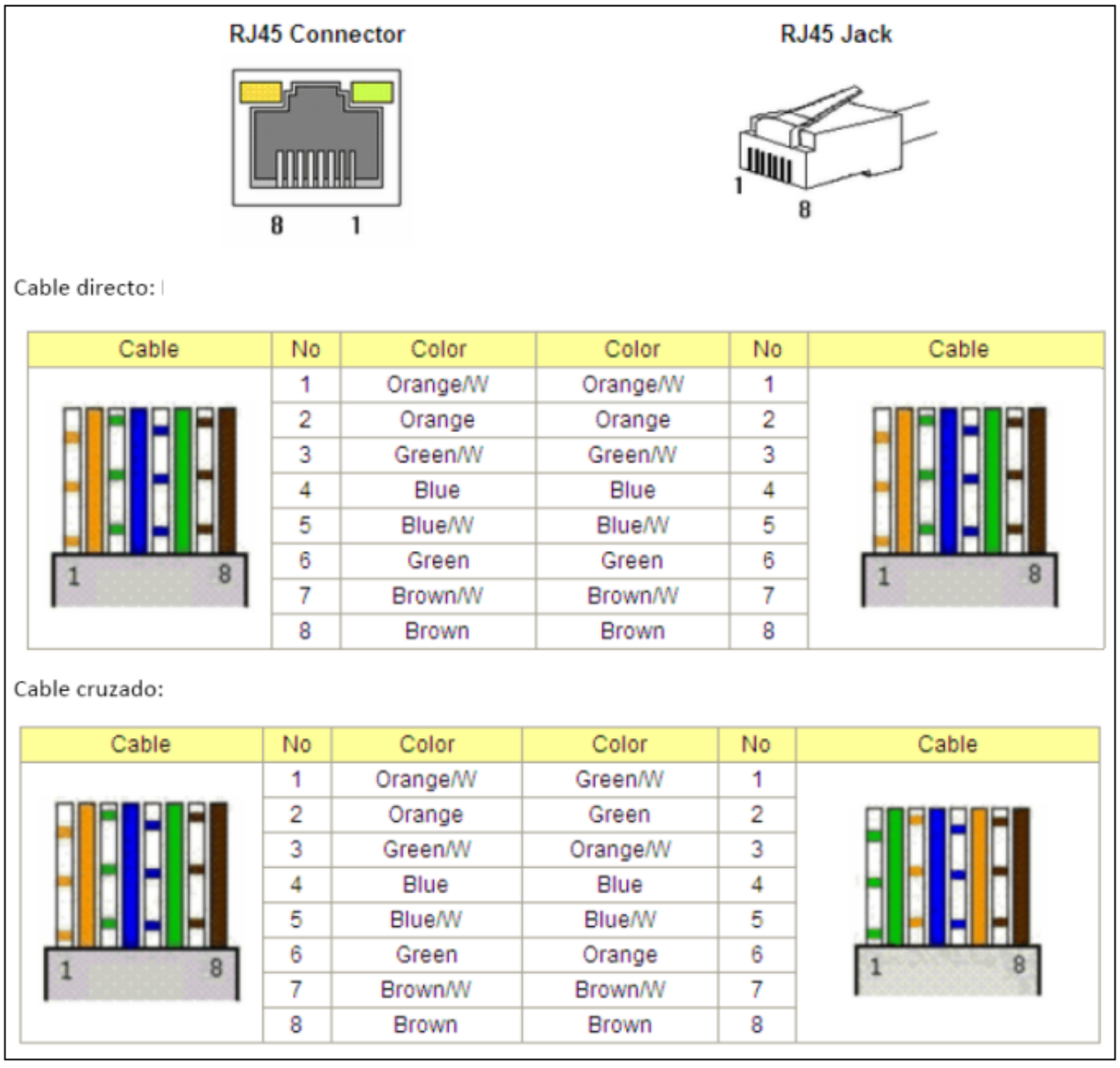


Figura 93: conexión entre PLCs y Dispositivo remotos via Rs485.

11.1.2 Conexiones sobre el puerto Ethernet:

Para La conexión del PLC dentro de una red Ethernet se utiliza los cables UTP conocidos ya sea cruzado o directa cuando se utiliza un Switch Ethernet como intermediario.

Cuando se conecta un HMI Cimon y un PLCs Cimon de manera directa se tiene que utilizar un cable Cruzado ethernet.



Cable directo:

Cable	No	Color	Color	No	Cable
	1	Orange/W	Orange/W	1	
	2	Orange	Orange	2	
	3	Green/W	Green/W	3	
	4	Blue	Blue	4	
	5	Blue/W	Blue/W	5	
	6	Green	Green	6	
	7	Brown/W	Brown/W	7	
	8	Brown	Brown	8	

Cable cruzado:

Cable	No	Color	Color	No	Cable
	1	Orange/W	Green/W	1	
	2	Orange	Green	2	
	3	Green/W	Orange/W	3	
	4	Blue	Blue	4	
	5	Blue/W	Blue/W	5	
	6	Green	Orange	6	
	7	Brown/W	Brown/W	7	
	8	Brown	Brown	8	

Figura 94: Pines puerto Ethernet.

11.1.3 Conexiones sobre el puerto USB

El puerto USB disponible sobre el PLC Cimon, es para poder establecer comunicación con una PC, para poder descargar y leer los programas sobre el PLC, asi tambien como para actualizar el Firmaware del PLC. Se utiliza un cable USB-Mini Usb.

11.2 Conexión de las entradas y salidas Digitales

dentro del PLCs se incluuen según el modelo un numero determinado de E/S, para poder la forma de conectarse es necesario siempre ver el manual que viene con el dispositivo hardware.

Para el caso del PLCs CM3-SP32MDT, este posee 16 entradas y 16 salidas digitales, la disposicion de sus terminales y conexión es a traves de la bornera CMO-TB32.

Tambien es posible utilizar otro accesorio de conexión, pero este debe tener la misma disposicion de los cables a si como ser adaptable a la conexión del terminal del PLC o los modulos.

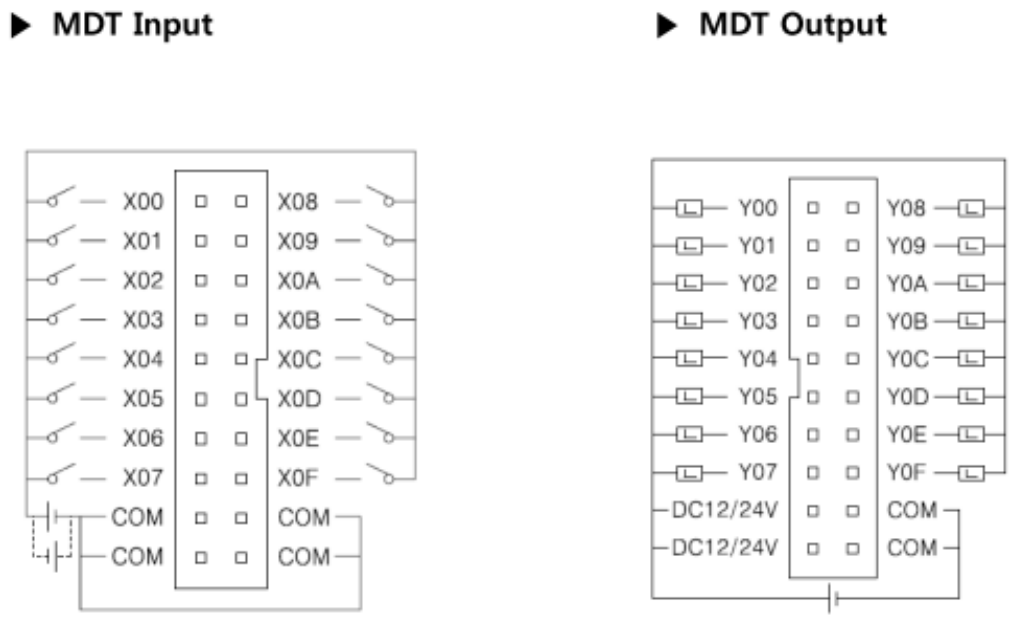


Figura 95: Conexionado sobre E/S Modulo digital PLC

- Todos los terminales COM del MDT Input, están conectados interiormente, y no es necesario unirlos externamente.
- Los terminales de salida y entrada son independientes entre si.
- Los terminales DC12/24V y COM en los terminales de salidas están disponibles en 2 terminales cada uno. (interconectados entre si).

Cuando se utiliza la bornera CMO-TB32, es necesario tener en cuenta a que modulo de expansion es conectado, la siguiente configuracion se representa según el modelo de PLC CM3-SP32MDT.

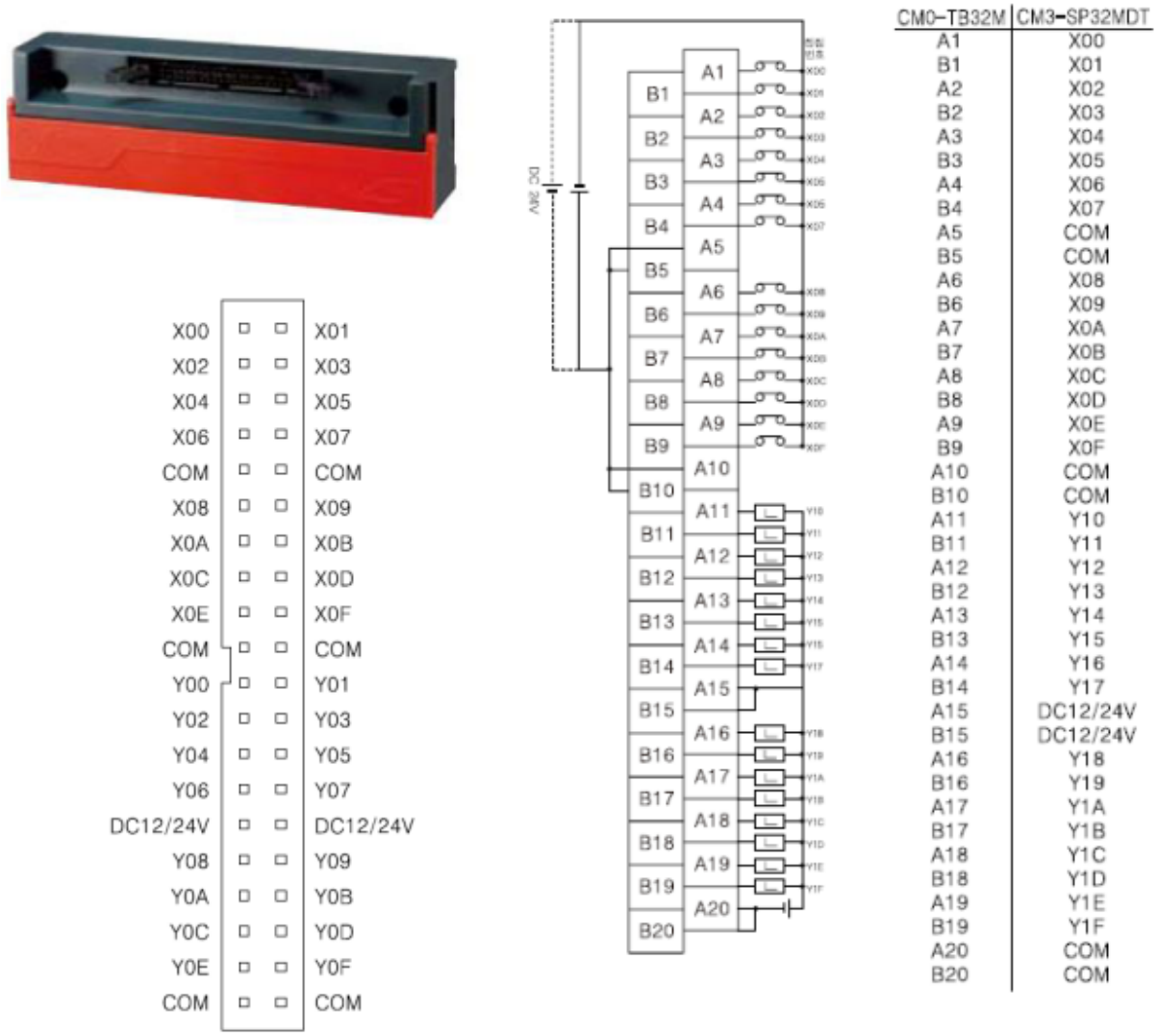


Figura 96: Conexionado sobre Bornera CMo-TB32M

Para conectar alguna entrada se puede usar solo algunos de los terminales COM.

Según el tipo de modulo, la conexión hardware de salida puede ser del tipo, Fuente, sumidero o Rele.

Para esos casos la conexión se realiza según el diagrama mostrado en la figura 97.

Ahora en funcion del modulo, a conectar la disposicion de los terminales cambia, en la figuras 98, 99 se ve la distintas configuracion según el modelo de equipo

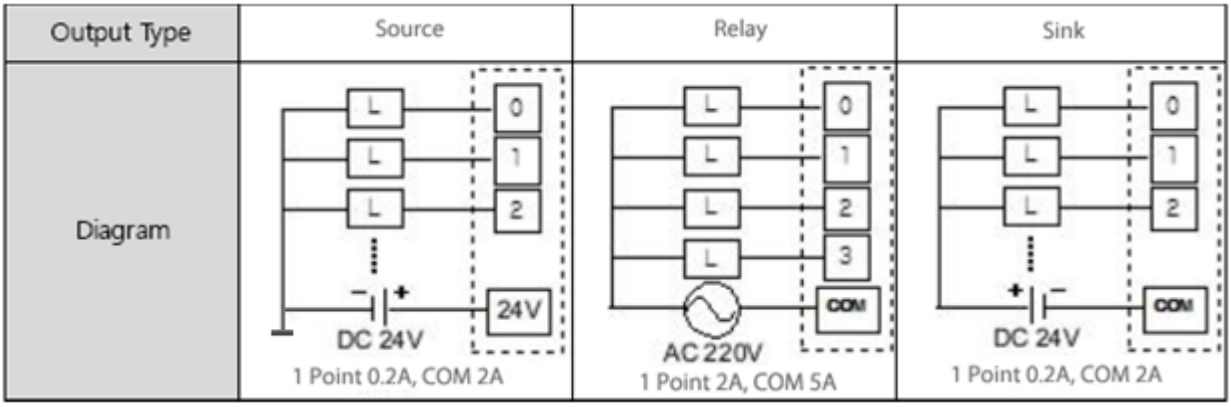


Figura 97: Conexionado de salidas del PLC

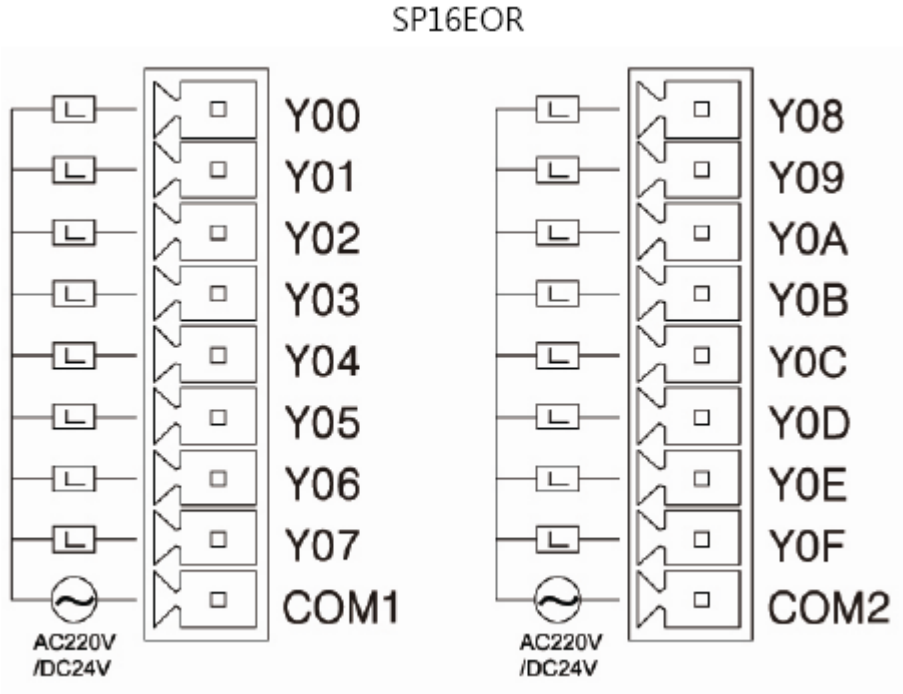


Figura 98: Conexionado de salidas para modulo tipo Rele

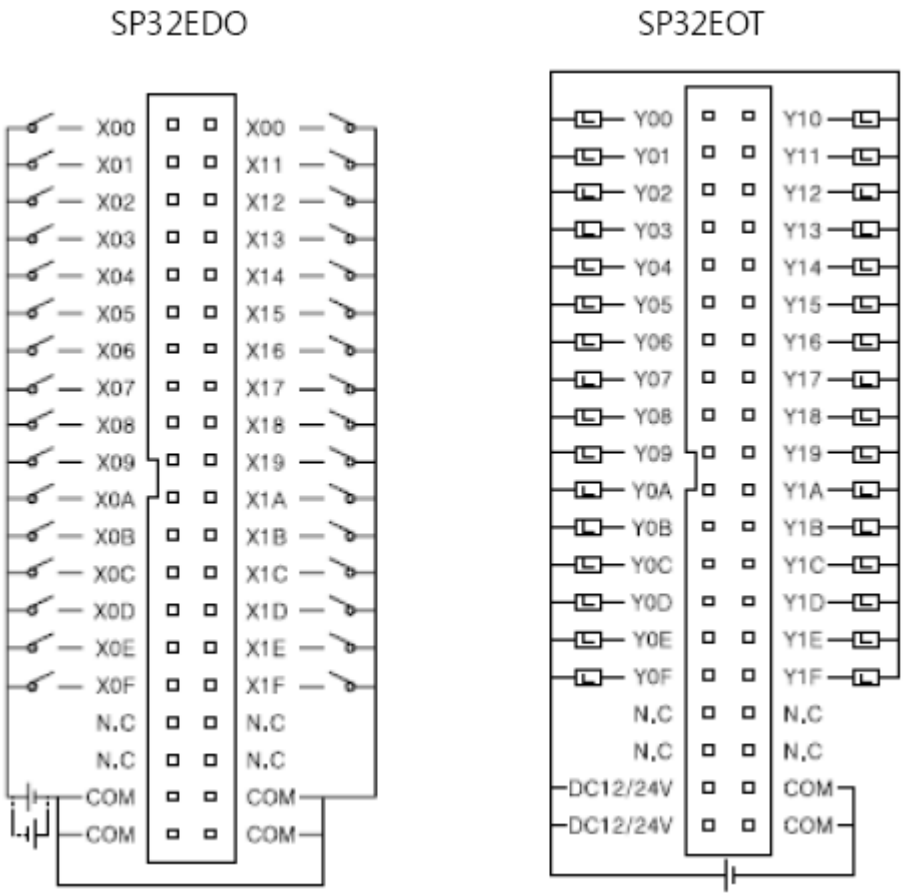


Figura 99: Conexionado de salidas para modulos digitales SP32ED0 y SP32EOT

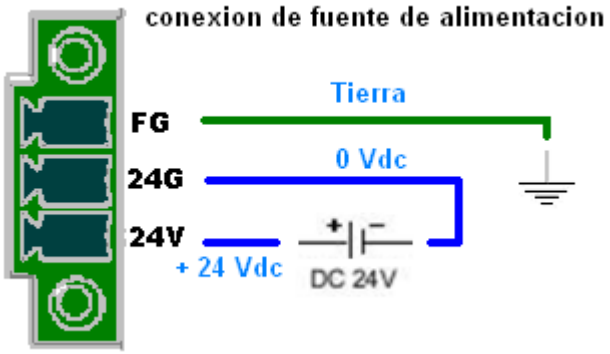


Figura 100: Conexionado de la fuente de alimentacion.

11.3 Conexión de las entradas y salidas Analogas

La conexión para estos tipos de modulos viene graficada en la siguiente figura, esto es según el modelo del equipo

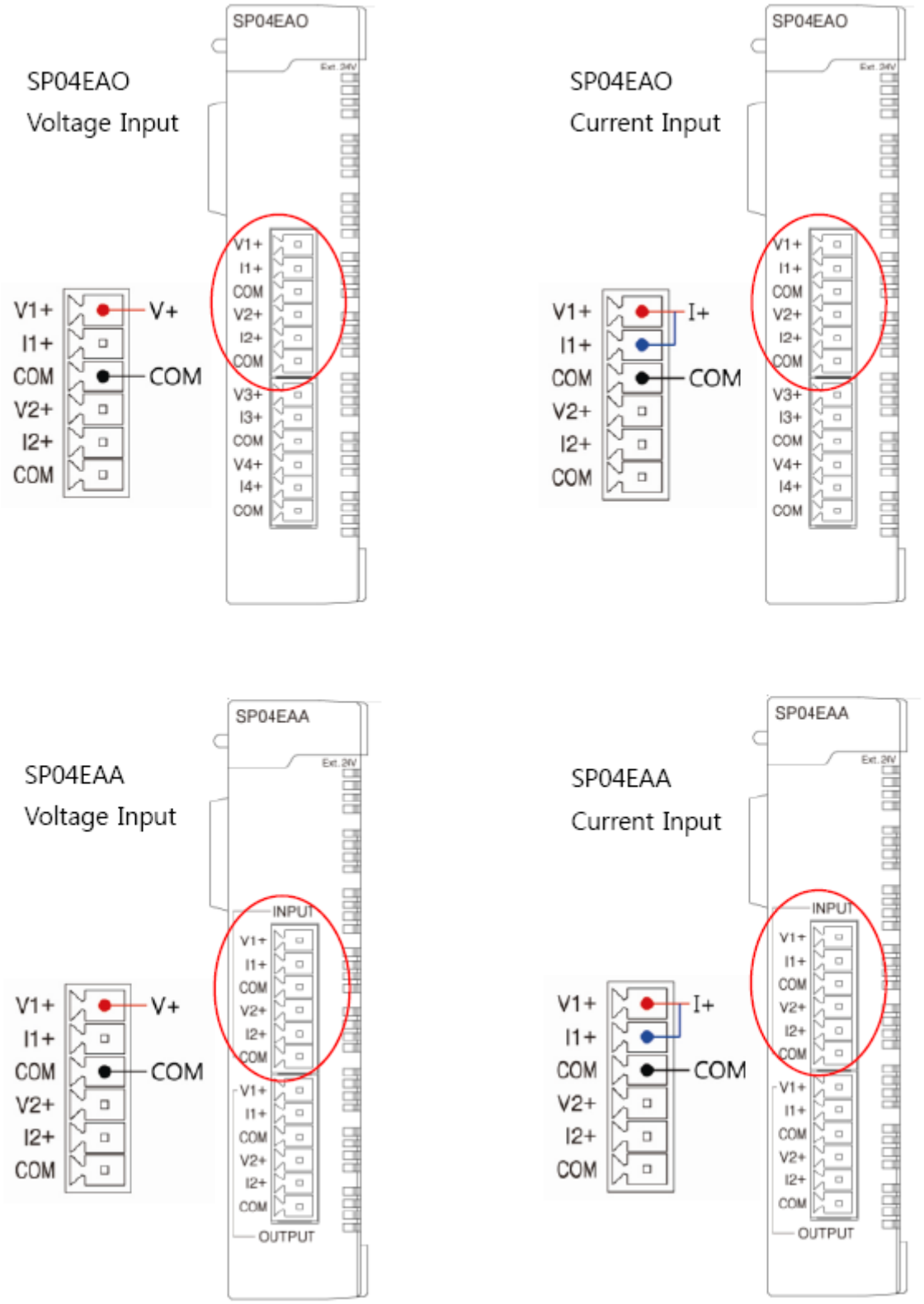


Figura 101: Conexionado de entradas analogas según el modelo de equipo.

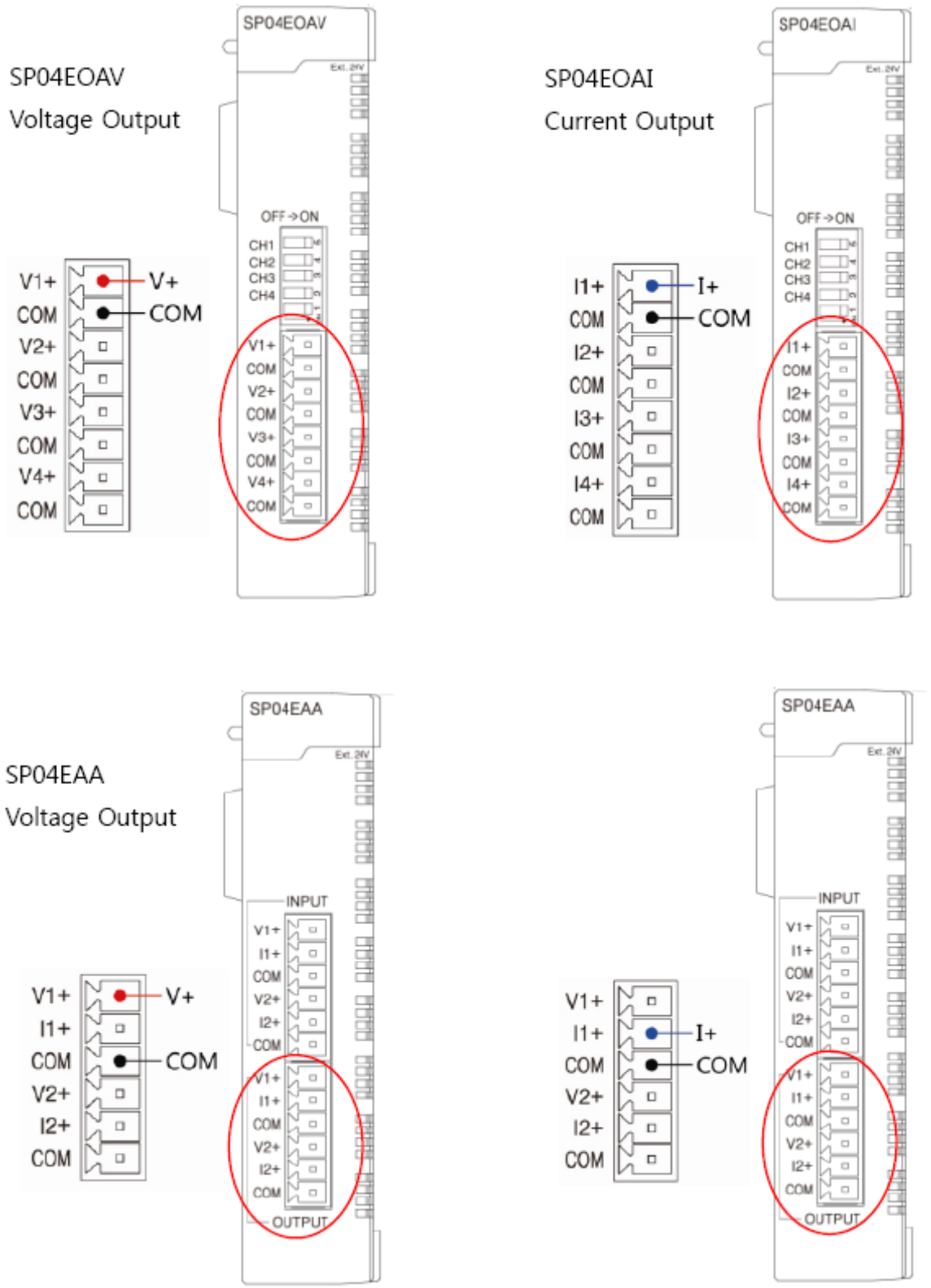


Figura 102: Conexionado de salidas analogas según el modelo de equipo.

11.4 Especificaciones tecnicas

Entradas y salidas digitales

Items	DC Input	Relay Output	TR Output
Rated I/O Voltage	DC 24V	AC 220V / DC 24V	DC 12V / 24V
Rated I/O Current	4mA	1point 2A / COM 5A	1point 0.2A, COM 2A
On Voltage/Current	DC19V / 3mA	-	-
Off Voltage/Current	DC6V / 1mA	-	-
Response Time	3ms or less	10ms or less	1ms or less
Operation Indication	LED will be ON when the Input is ON		
Insulation Type	Photo coupler Insulation	Relay Insulation	Photo coupler Insulation
Input Type	SINK / SRC	-	-

Entradas analogas:

Item		CM3-SP04EAO	CM3-SP04EAA
Analog Input point		4pts.	2pts.
Analog Input		Voltage : 0V ~ +5V, 1V ~ +5V, 0V ~ +10V. -10V ~ +10V Current : 0~20mA, 4~20mA	
Resolution		14 bit, 0~16000	14bit, 0~16000 16bit, 0~64000 Setting by CICON
Accuracy		± 0.1% (Full Scale)	
Conversion speed		2.1ms / 1ch	
Absolute Max. input		Voltage: ±12V Current: ±25mA	±12V ±25mA
Insulation		Photo-coupler insulation between input terminal and PLC, (No insulation between channels)	
I/O points occupied		16pts	
Power supply		External DC 24V	
Connection Terminal		12pts. Terminal Block	
Consumption current (mA)	Inner	@DC24V, 15mA	@DC24V, 15mA
	External	@DC24V, 60mA	@DC24V, 70mA

Salidas analogas:

Item		SP04EAA	SP04EOAV	SP04EOAI
Analog output point		2 pts.	4 pts.	4 pts.
Analog output		Voltage : 0V ~ +5V, 1V ~ +5V, 0V ~ +10V. -10V ~ +10V Current : 0~20mA, 4~20mA	0V ~ +10V. -10V~ 10V (switch mode)	4 ~ 20mA
Output setting method		Setting by CICON	Switch OFF : 0V ~ 10V Switch ON : -10 ~ 10V CICON setting	Not possible
Digital Input		14bit, 0~16000 16bit, 0~64000 Setting by CICON	14bit, 0~16000	
Accuracy		± 0.1% (Full Scale)		
Conversion velocity		10ms		
Absolute Max. Output		±12V 24mA	±12V	24mA
Insulation method		Photo Coupler between Input terminal and PLC		
Power supply		External DC 24V		
I/O occupation		16 pts.		
I/O Terminal		12pts Terminal		
Consumption current (mA)	Internal	@DC24V, 15mA	@DC24V, 15mA	@DC24V, 15mA
	External	@DC24V, 70mA	@DC24V, 60mA	@DC24V, 70mA

Performance PLCs

Items		Specifications
Power		DC24V
Program Control Method		Stored Program, Process-Driven Interrupt, Time Driven Interrupt
I/O Control Method		Indirect method, Directed by program instruction
Program language		IL(Instruction List), LD(Ladder Diagram)
Data Processing Method		32 Bit
Instructions	Sequence	55 Instruction
	Application	389 Instruction
Processing speed (Sequence)		200ns / Step
Program capacity		10K Step
Max. I/O, Max. expansion		384 pts / 1 main Block + Max. 11 Block
Operation mode		Run, Stop, Remote Run, Remote Stop
Back-up method		Data which set up as retain
Total program		128
Program types	Scan	(127-Program block)
	Cyclic task	16
	Special	126
	Initial task	2 (_INIT,_H_INIT)
	Subroutine	126
Self-diagnosis		Detects errors of scan time, memory, I/O and power supply, battery error
Re-start		Cold, Hot re-start
Device memory capacity	X	1024 pts (X0000 - X063F)
	Y	1024 pts (Y0000 - Y063F)
	M	8192 pts (M0000 - M511F)
	L	4096 pts (L0000 - L255F)
	K	4096 pts (K0000 - K255F)
	F	2048 pts (F0000 - F127F)
	T	512 pts (T0000 - T0511)
	C	512 pts (C0000 - C0511)
	S	100 states x 100 set (00.00 - 99.99)
	D	10000 words (D0000 - D9999)
	Z	1024 words (Call Stack : Z0000 - Z0063, Z1000 - Z1063)
	R	16 pts (Index)
High Speed Counter		20Kpps, 2 Phase 2Ch.
Positioning		Max. 100Kpps, 2 Axis, Linear interpolation
PID		32 Channels, Auto-Tuning
RTC		Embedded (Battery CR2032 Backup)
Comm. Channel		Standard : USB Loader, Serial 1(RS232C) Optional : Serial 1Ch(RS485), Ethernet 1Ch
Etc.		